HEI: Hunter Events Interface
A platform based on services for the detection and reasoning about events

Antonio Sorgente, Antonio Calabrese, Gianluca Coda, Paolo Vanacore and Francesco Mele
Institute of Applied Sciences and Intelligent Systems "Eduardo Caianiello"
National Research Council
Via Campi Flegrei 34, 80078 Pozzuoli (Naples) Italy
{a.sorgente, a.calabrese, g.coda, p.vanacore, f.mele}@isasi.cnr.it

Abstract

In this paper we present the definition and implementation of the Hunter Events Interface (HEI) System. The HEI System is a system for events annotation and temporal reasoning in Natural Language Texts and media, mainly oriented to texts of historical and cultural contents available on the Web. In this work we assume that events are defined through various components: actions, participants, locations, and occurrence intervals. The HEI system, through independent services, locates (annotates) the various components, and successively associates them to a specific event. The objective of this work is to build a system integrating services for the identification of events, the discovery of their connections, and the evaluation of their consistency. We believe this interface is useful to develop applications that use the notion of story, to integrate data of digital cultural archives, and to build systems of fruition in the same field. The HEI system has been partially developed within the TrasTest project [1].

1 Introduction

The amount of digital resources is growing day by day, and this raises new challenges in the management process in particular in the access and reuse of such resources. For this aim we need systems that offer solutions to effectively store digital resources (text, images, video, etc.) and allow us to give them an interpretation with respect to a given semantic.

Our contribution in this direction regards the definition and implementation of a system that permits a user to manage resources (mainly textual resources, but also media resource) and easily annotate them. We called this system HEI (Hunter Events Interface).

Today in research areas of Natural Language (NL) and temporal reasoning there are numerous tools that help NL text analysis and facilitate applications in many different sectors. Many research groups have made available on the Web some of their own developed tools (parsers, name detectors, etc ..) providing, in particular in the field of temporal representation and reasoning, a little support for the integration of such services. This work is born with the goal of developing a methodology that integrates services concerning annotation, discovery, connectivity, and temporal reasoning through events in natural language texts provided by different developers and also, offering services to annotate media. Through the use of such annotations, we can define multimedia streams that coherently synchronises media elements with a synthetic voice delivering the textual content, and retrieve information to respond to queries as in dialog systems.

For the annotation of natural language texts we have chosen CSWL (Cultural Story Web Language) as referred semantic formalism, and according to CSWL we have established the following annotation approach, that can be made by the particular HEI architecture:

1. for each text in NL, it applies n basic services Sn automatically annotating the text with significant labels for the CSWL semantics (basic services Sn are available in Web or are part of an internal repository HEI) producing m annotated entities (see section 4.1);

[1] TrasTest (Transforming Texts on Cultural Knowledge in Structured Data) is a project of National Research Council (CNR)

Proceedings of the Workshop on Events and Stories in the News, pages 79–88
Santa Fe, New Mexico, USA, August 20, 2018.

2. some services (algorithms on temporal grammatical, semantic or pragmatic properties emerging from the NL text) are applied to extend the m entity previously annotated (see section 4.2);
3. we can monitor the made annotations, and interactively execute the completion of annotations.

In literature there are some systems that allow browsing and text annotation (Hou et al., 2015; de Boer et al., 2015; Vossen et al., 2016). In (Hou et al., 2015; Vossen et al., 2016) the authors presented NewsMiner and NewsReader system respectively, and both are focused on the interpretation of news using an event-based formalism. In (de Boer et al., 2015), in particular, the focus is on historical events. Our system is also based on an event formalism, but in addition to events takes into account fluents and complex events. Moreover HEI system allows users to evaluate the temporal consistency of text, and can be expanded with the addition of new annotation services.

In this paper we introduce the CSWL formalism (section 2), the architecture of the HEI system (section 3) and some services implemented in the system (section 4).

2 CSWL Formalism

CSWL (Cultural Story Web Language) is a formalism based on a cognitive notion of event, based on five ”Ws” (Who, What, Where, When and Why) (Sorgente et al., 2016a; Mele and Sorgente, 2015). We consider an event as an action that happens over time that has participants and a location where it takes place. Differently from other event-based formalism (such as those defined in (van Hage et al., 2011; Wang et al., 2007; Scherp et al., 2009)), through CSWL formalism we can also represent the major entities that change over time: properties of event participants, spatial and meronomic relations of objects composing the architectural structures, and function of artifacts. We represent such entities as fluents, the same concept defined in the Event Calculus (Mueller, 2015). In the same way, in CSWL social roles of people (or historical characters) are entities that change over time, and are represented as relationships representing capabilities to perform potential actions. We have inserted in the formalism primitives to represent emotional roles owned by artworks in relation to their shapes and colours. With CSWL we can represent complex events as single units (aggregations of more events), having each a name and properties set. In CWSL, stories are complex events represented by a set of events having causal (and therefore temporal) relations between them. In CSWL formalism we can represent concepts based on events through classes, on which we can associate (and apply) inference rules. All the entities that change over time – the properties, roles, spatial relationships, and mental attitudes of the characters – are grouped together in CSWL in a same class (fluent), and rules for the reasoning on persistence and change of properties over time, are applied uniformly to each element of such class.

We report in the form of first order logic relations, the main classes of CSWL. In the ontological formulation of CSWL, there are three basic entities:

- events (simple) - are entities that happen on a time interval;
- fluents - are entities that change over time, they are true (or false) at specific instants (or time intervals), can cease or start to be true again;
- complex events - are aggregations of simple events or other complex events

```
% AnythingInTime Taxonomy
anythingInTime(X):-event(X). anythingInTime(X):-complexEvent(X).
anythingInTime(X):-fluent(X).
%Same categories of events
event(X):-reportingThingEvent(X). event(X):-state(X).
event(X):-occurrence(X).     event(X):-perceptionThingEvent(X).
%Event -- Definition Scheme
event(Ex):- hasWhen(Ex,Wnx), hasWhat(Ex,Wtx), hasWhere(Ex,Wrx),
   hasWho(Ex,Wox), wtRel(Wtx), woRel(Wox), wrRel(Wrx), whRel(Wnx).
whRel(Wnx):- has(Wnx,Tmodx), has(Tmodx,Intx), tmod(Tmodx), int(Intx).
tmod(Tmodx):-on(Tmodx). tmod(Tmodx):-after(Tmodx). tmod(Tmodx):-prec(Tmodx).
tmod(Tmodx):-in(Tmodx). tmod(Tmodx):-finish(Tmodx). tmod(Tmodx):-start(Tmodx).
%Complex event -- scheme definition
complexEvent(Cex):- hasWhenC(Cex,Cwn), hasWhatC(Cex,Cwt),
                 hasWhereC(Cex,Cwr), hasWhoC(Cex,Cwo).
%Property Fluent -- scheme definition
```

```
fluent(F):- hasWhenF(F, WhenF), hasWhatF(F, WhatF),
            relWhenF(WhenF), relWhatF(WhatF).
relWhenF(WhenF):-temporalTerm(WhenF).
relWhenF(WhenF):-int(WhenF). relWhenF(WhenF):-after(WhenF).
relWhenF(WhenF):-prec(WhenF). relWhenF(WhenF):-finish(WhenF).
relWhenF(WhenF):-start(WhenF). relWhenF(WhenF):-in(WhenF).
relWhat(WhatF):-property(WhatF). relWhat(WhatF):-role(WhatF).
relWhat(WhatF):-mentalAct(WhatF). relWhat(WhatF):-partPartWhoRel(WhatF).
relWhat(WhatF):-spatialRelation(WhatF).
relWhat(WhatF):-emotionalFunction(WhatF). relWhat(WhatF):-artFunction(WhatF).
role(WhatF):-socialRole(WhatF).
%Property -- scheme definition
property(Propx):- hasPrName(Propx, Pnamex),
   hasPrSubject(Propx, Subjx), hasPrValue(Propx, Valuex).
%Role - definition
role(Rox):- hasName(Rox, RoleNamex),
   hasRoParticipant1(Rox, Px1), hasRoParticipant2(Rox, Px2).
%Mental Act
mentalAct(Mactx):- hasMentalAtt(Mactx,Mattx), hasAgent(Mattx,Agx),
   hasEvent(Mattx,Ex), mentalAtt(Mattx), event(Ex), agent(Agx).
% Types of Mental Attitudes
mentalAtt(X):-believe(X). mentalAtt(X):-intention(X). mentalAtt(X):-desire(X).
%Meronomic Relation
partPartWhoRel(PartWhoRelx):- hasType(PartWhoRelx,TypeRelPx),
   hasPart(TypeRelPx,Partx), hasWhole(TypeRelPx, Wholex), typeRelPt(TypeRelPx).
% Types of Meronomic Relations
typeRelPt(X):-relPtCo(X). % Component-Object.
typeRelPt(X):-relPtMc(X). % Member - Collection.
typeRelPm(X):-relPtMc(X). % Portion - Mass.
typeRelMo(X):-relPtMc(X). % Material - Object.
typeRelCa(X):-relPtMc(X). % Feature - Activity.
typeRelSa(X):-relPtMc(X). % Site - Area.
%A taxonomy of Spatial Relations
spatialRelationType(Sr):-qualitativeRel(Sr).
spatialRelationType(Sr):-quantitativeRel(Sr).
quantitativeRel(Sr):- polar(Sr). quantitativeRel(Sr):- logitudinal(Sr).
quantitativeRel(Sr):-cartesian(Sr). qualitativeRel(Sr):-directional(Sr).
qualitativeRel(Sr):- proximity(Sr). qualitativeRel(Sr):- topological(Sr).
topological(Sr):- contains(Sr). topological(Sr):- covers(Sr).
topological(Sr):- meets(Sr). topological(Sr):- overlying(Sr).
topological(Sr):- covered(Sr). topological(Sr):- overlap(Sr).
directional(Sr):- over(Sr). directional(Sr):- high(Sr).
directional(Sr):- under(Sr). directional(Sr):- cardinal(Sr).
directional(Sr):- low(Sr). directional(Sr):- right(Sr).
directional(Sr):- left(Sr). cardinal(Sr):- east(Sr).
cardinal(Sr):- south(Sr). cardinal(Sr):- north(Sr). cardinal(Sr):- west(Sr).
%Social Role
socialRole(RolRelx):- hasName(RolRelx,Namex), hasRoParticipant1(RolRelx, Wo1),
   hasRoParticipant2(RolRelx, Wo2), hasCapability(Wo1,Canx),
   capability(Canx), participant(Wo1), participant(Wo2).
capability(Canx):- hasAction(Canx,Actionx), action(Actionx).
%Artifact Function
artificat(Artx):- hasStructur(Artx,Strx), hasFunction(Artx,Funx),
   hasBehavior(Artx,Behx), artStructur(Strx),
   artFunction(Funx), artBehavior(Behx).
artStructur(Strx):- hasSpatialRelation(Strx,Relx), spatialRelation(Relx).
spatialRelation(Relx):- hasType(Relx,TypeRelPx),spatialRelationType(TypeRelPx),
   hasComponent1(Relx, C1), hasComponent2(Relx, C2),
   artifactPart(C1), artifactPart(C2).
artFunction(Funx):-
   hasName(Funx,Namex), hasCapability(Funx,Canx), capability(Canx).
artBehavior(Behx):- hasName(Behx,Namex), hasStimulus(Behx ,Stimulusx),
   hasEffect(Behx,Effectx), cause(Stimulusx, Effectx),
   event(Stimulusx), event(Effectx).
%Emotional Function
emotionalFunction(EmRelx):- hasName(EmRelx,Namex), hasEmObject(EmRelx, Objx),
   hasEmParticipant(EmRelx, Wox), hasCapabilityEm(Objx,Canx), capabilityEm(Canx),
   participant(Objx),participant(Wox).
addEmot(bel(Wox, Emotionx)):- visuaAct(Wox,Objx), emotionalFunction(EmRelx),
   hasEmObject(EmRelx, Objx), hasEmParticipant(EmRelx, Wox), emotion(Emotionx).
%A taxonomy for Natural Entities
naturalEntity(X):-socialAgent(X). naturalEntity(X):-naturalPhenomenon(X).
naturalEntity(X):-physicalAgent(X). naturalEntity(X):-physicalObject(X).
physicalAgent(X):-person(X).  physicalAgent(X):-animal(X).
socialAgent(X):-groupAgent(X). socialAgent(X):-collectiveAgent(X).
physicalObject(X):-media(X).  physicalObject(X):-artifactPart(X).
%Temporal expression classes
```

```
temporalExp(X):-interval(X).  temporalExp(X):-temporalTerm(X).
temporalTerm(X):-dataValue(X). temporalTerm(X):-timeValue(X).
dataValue(X):-dateCalendar(X). dataValue(X):-dateWeek(X).
dataValue(X):-dataQualitative(X). timeValue(X):-timeQualitative(X).
timeValue(X):-clock(X).
% Definitions for the temporal taxonony
clock(Clx):-
   hasHour(Clx, Hourx), hasMinute(Clx, Minutex), hasSecond(Clx, Secondx).
date(Dcx):- hasYear(Dcx, Yearx), hasMonth(Dcx, Monthx), hasDay(Dcx, Dayx).
dataCalendar(Dayx,Monthx,Yearx):- date(Dcx), hasYear(Dcx, Yearx),
   hasMonth(Dcx, Monthx), hasDay(Dcx, Dayx).
dateWeek(Dwx):- hasMonth(Dwx, Monthx), hasDay(Dwx, Dayx).
dateWeekCalendar(Monthx,Dayx):-
   dateWeek(Dwx), hasMonth(Dwx, Monthx), hasDay(Dwx, Dayx).
interval(Intx):-
   begin(Intx,Tt1), end(Intx, Tt2), temporalTerm(Tt1), temporalTerm(Tt2).
dataQualitative(Dq):- hasTimeQualitative(Dq,Tq),
   convertFunction(Tq,Datax), timeQualitative(Tq), dataValue(Datax).
timeQualitative(Tq):- annotation(Tkx, Tq), token(Tkx).
```

3 Architecture

The HEI [2] (Hunter Events Interface) system consists of software modules which assist the user in the semantic annotation process of NL texts and multimedia containing events. The HEI framework is based on a plugin architecture, so the system is modular and easily extendable. The plugin architecture (Fig. 1) allows us to extend the tool functionalities with respect to the format of the reference ontology, automatic annotation, reasoning on events, annotation and semantic multimedia mashup export. All the services shown below permit users to annotate automatically the text, but also, through the interface they can modify, add or remove the annotations.

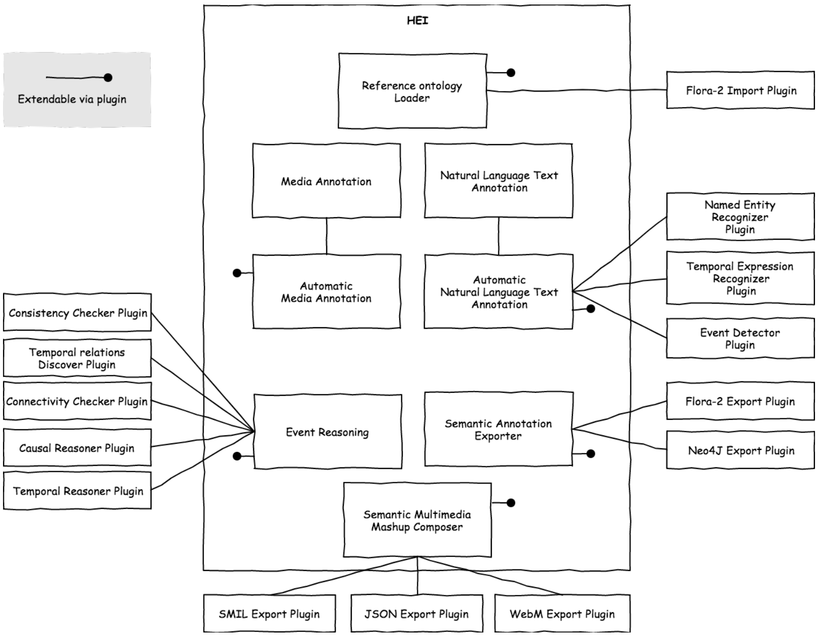

HEI
Extendable via plugin
Reference ontology Loader
Flora-2 Import Plugin
Media Annotation
Natural Language Text Annotation
Named Entity Recognizer Plugin
Automatic Media Annotation
Automatic Natural Language Text Annotation
Temporal Expression Recognizer Plugin
Event Detector Plugin
Consistency Checker Plugin
Temporal relations Discover Plugin
Connectivity Checker Plugin
Causal Reasoner Plugin
Temporal Reasoner Plugin
Event Reasoning
Semantic Annotation Exporter
Flora-2 Export Plugin
Neo4J Export Plugin
Semantic Multimedia Mashup Composer
SMIL Export Plugin
JSON Export Plugin
WebM Export Plugin

Figure 1: HEI Architecture diagram

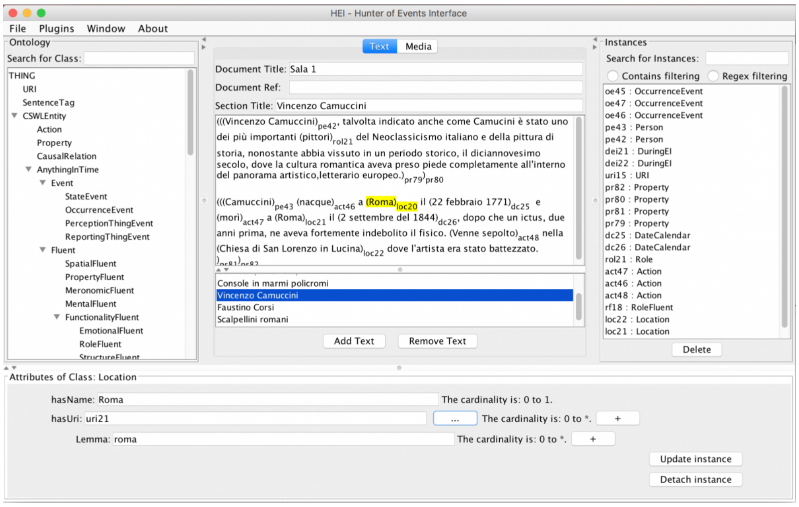
The particular architecture of HEI allows us to use different reference ontologies for the semantic annotation, and, through its *import plugins*, the system is able to import ontologies, written in different formalism. About this, we have developed a specific plugin "F2ImportPlugin" to load reference ontologies written in Flora-2[3].

In Fig. 2 a screen-shot of HEI interface is reported. On the left of the screen there is the reference ontology used to annotate the text, the latter is shown in the centre of the screen. The user selects the

[2] http://smcm.isasi.cnr.it/lab/hei-hunter-of-event-interface/

[3] http://flora.sourceforge.net/

class to be instantiated, from the left column, he can select the text to annotate, and in the bottom of the screen can see the instance attributes filled. On the right side of screen, the ontology instances are shown.



Document Title: Sala 1

Document Ref:

Section Title: Vincenzo Camuccini

(((Vincenzo Camuccini)$_{pe42}$, talvolta indicato anche come Camucini è stato uno dei più importanti (pittori)$_{rol21}$ del Neoclassicismo italiano e della pittura di storia, nonostante abbia vissuto in un periodo storico, il diciannovesimo secolo, dove la cultura romantica aveva preso piede completamente all'interno del panorama artistico,letterario europeo.)$_{pr79}$)$_{pr80}$

(((Camuccini)$_{pe43}$ (nacque)$_{act46}$ a (Roma)$_{loc20}$ il (22 febbraio 1771)$_{dc25}$ e (morì)$_{act47}$ a (Roma)$_{loc21}$ il (2 settembre del 1844)$_{dc26}$, dopo che un ictus, due anni prima, ne aveva fortemente indebolito il fisico. (Venne sepolto)$_{act48}$ nella (Chiesa di San Lorenzo in Lucina)$_{loc22}$ dove l'artista era stato battezzato.)$_{pr81}$)$_{pr82}$

Console in marmi policromi
Vincenzo Camuccini
Faustino Corsi
Scalpellini romani

oe45 : OccurrenceEvent
oe47 : OccurrenceEvent
oe46 : OccurrenceEvent
pe43 : Person
pe42 : Person
dei21 : DuringEI
dei22 : DuringEI
uri15 : URI
pr82 : Property
pr80 : Property
pr81 : Property
pr79 : Property
dc25 : DateCalendar
dc26 : DateCalendar
rol21 : Role
act47 : Action
act46 : Action
act48 : Action
rf18 : RoleFluent
loc22 : Location
loc21 : Location

Attributes of Class: Location

hasName: Roma The cardinality is: 0 to 1.

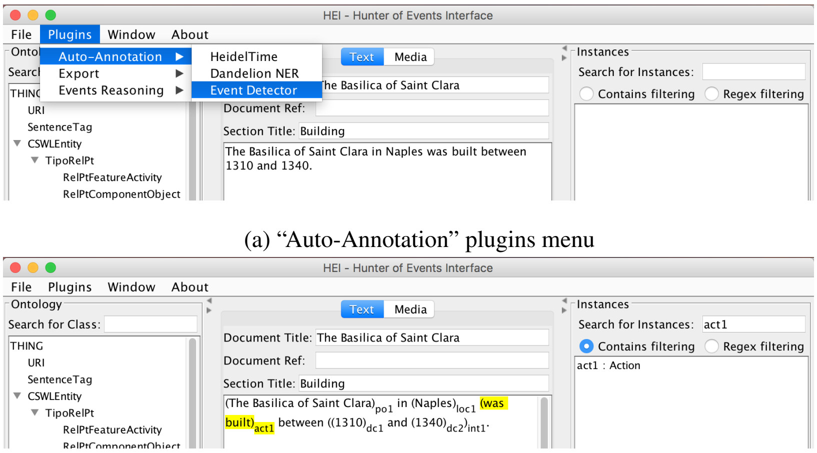
hasUri: uri21 ... The cardinality is: 0 to *. +

Lemma: roma The cardinality is: 0 to *. +

Figure 2: HEI Annotation GUI

The main aim of HEI is to have an environment that permits in automatic or assisted way, the annotation of temporal entities, and the analysation of the text on such annotations. For this reason a set of plugins has been built. Given a text in NL, with the HEI interface we access to the "$Plugins \rightarrow Auto - Annotation$" menu (Fig. 3a), and can launch the plugins for automatic text annotation (Fig. 3b).

Developed plugins are based on services described in 4.1. We have implemented: "DandelionPlugin" for name entity recognition using the Dandelion API[4] to detect the participants in events; "EventDetectorPlugin" for detecting events; and "TemporalExpressionPlugin" using HeidelTime(Strötgen and Gertz, 2015), for annotating temporal expressions. Plugins architecture of the HEI system adopts the approach described in (Vanacore et al., 2016), where a method for defining mapping between different representations has been presented. In order to match instances of the representation models adopted by the auto-annotation services, with the reference ontology used for the annotation, we use an interface for executing the mapping among different representations. For this aim we use some bridge rules allowing the semantic linking of the entities (see Fig. 4.)



he Basilica of Saint Clara

Document Ref:

Section Title: Building

The Basilica of Saint Clara in Naples was built between 1310 and 1340.

(a) "Auto-Annotation" plugins menu

Document Title: The Basilica of Saint Clara

Document Ref:

Section Title: Building

(The Basilica of Saint Clara)$_{po1}$ in (Naples)$_{loc1}$ (was built)$_{act1}$ between ((1310)$_{dc1}$ and (1340)$_{dc2}$)$_{int1}$.

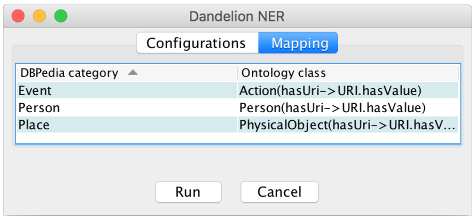
Search for Instances: act1

act1 : Action

(b) example of automatic detected events by the "EventDetector" plugin

Figure 3: HEI Annotation GUI

[4] https://dandelion.eu



DBPedia category ▲	Ontology class
Event	Action(hasUri->URI.hasValue)
Person	Person(hasUri->URI.hasValue)
Place	PhysicalObject(hasUri->URI.hasV...

Figure 4: HEI GUI - Example of mapping rules for the Dandelion NER service

The HEI architecture allows us to add reasoning services, mapping concepts present in referring ontology with the schema adopted by the reasoner. In this case, after the process of annotation, the user can run reasoning services using the implemented mapping. At this time of HEI system development are present various services for events reasoning.

To store the annotation, we developed some plugins for annotations export: "F2Export Plugin" to export annotations in Flora-2, and "NEO4JExportPlugin" to export annotations into Neo4J[5] graph. In Fig. 5 we report a screenshot of a Neo4J graph, obtained by the "NEO4JExportPlugin", with 4860 nodes and 7468 relationships.

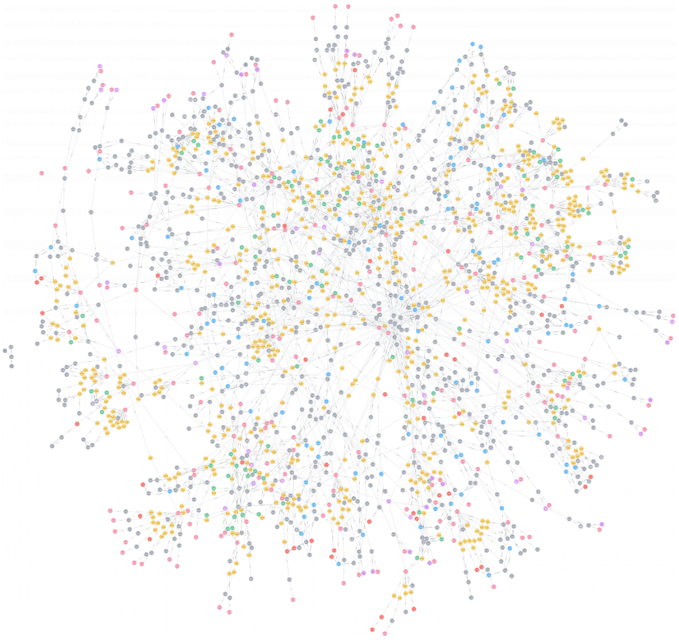


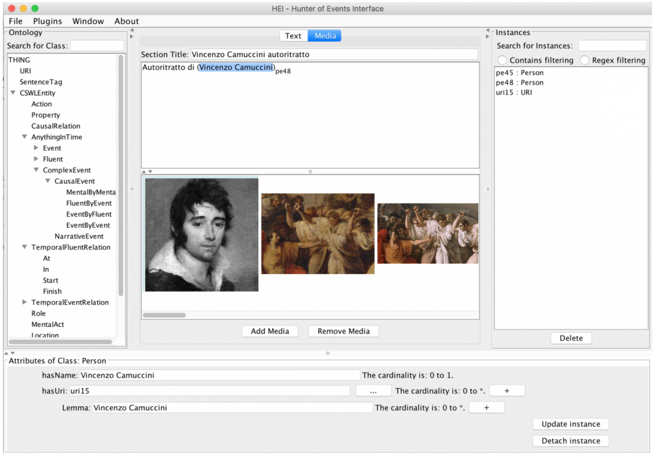
Figure 5: Example of Neo4J graph obtained with the "NEO4JExportPlugin"

Through HEI we can semantically annotate the media (Fig. 6). The annotator can adds video or images, specifying a remote Uniform Resource Locator (URL), or selecting a file from the local filesystem. The annotations can be associated to a text related to the media or to the media directly. Having different resource types annotated with the same semantic, we can define services that permit building of artefacts by composing such resources. Through this annotation, it is possible produce multimedia presentations that synchronise text, images and video in a unique stream. Developed plugins can export multimedia artefacts in SMIL[6] JSON and Webm[7] formats. An application of such services was presented in (Sorgente et al., 2016b)

[5] https://neo4j.com/

[6] https://www.w3.org/TR/SMIL/

[7] http://www.webmproject.org/



Section Title: Vincenzo Camuccini autoritratto

Autoritratto di (Vincenzo Camuccini)pe48

pe45 : Person
pe48 : Person
uri15 : URI

Attributes of Class: Person

hasName: Vincenzo Camuccini The cardinality is: 0 to 1.
hasUri: uri15 ... The cardinality is: 0 to *. +
Lemma: Vincenzo Camuccini The cardinality is: 0 to *. +

Figure 6: HEI Media Annotation GUI

4 Services

The CSWL formalism represents historical entities through classes of an ontology. This permits us to facilitate the automatic annotation processes, the reuse of the existing axiomatics in the research of temporal reasoning and the creation of new ones. In this section we present the HEI framework services for texts analysis. Some basic services regard the annotation process, they are used to identify the concepts inside the text and to represent them through the formalism defined in section 2, while other services of temporal reasoning extract new relations among temporal entities.

4.1 Annotation services

The services used in the HEI framework are known in the literature (such as Danedelion, HeidelTime) and services defined by us (such as EventDetection, TempRel).

Named Entity Recognition. Named Entity Recognition (NER) is a service that recognises locations, persons, and other entities inside the text. We have implemented NER plugin that uses Dandelion, a set of services offered by Spaziodati[8]. NER provides as output a set of expressions with associated semantic information. Often, some results do not have any associated semantic label, Dandelion API classifies them as a generic concept. For these reasons in HEI exists a taxonomy of lexical terms for cultural heritage domain that allows us to associate the right category to art objects: painting, sculpture etc..

Temporal Expression Recognition. Temporal Expression Recognition (TER) is a service that recognises temporal expressions in the text. For the implementation of *TER plugin* we have used HeidelTime (Strötgen and Gertz, 2015) that is a "multilingual, domain-sensitive temporal tagger" developed at the Database Systems Research Group at Heidelberg University. The service annotates temporal expressions, such as times, dates, durations, etc., in NL texts and normalises them according to the TIMEX3 tag defined in the TimeML Specification[9]. HeidelTime is based on hand-crafted resources for 13 languages, and starting from the version 2.0, contains automatically generated resources for over 200 languages. For each language HeidelTime is able to distinguish between news and narrative document types, in particular for English only, colloquial documents and scientific articles are also supported. The output formats of the service are TimeML and XMI.

Temporal Relation. The service allows us to annotate temporal relationships between events and temporal expressions within a text. For this purpose, besides the text, the service requires that events and temporal expressions have already been annotated. The latter are provided by HeidelTime and Event-Detector. The temporal relation service is based on rules defined on the dependency tree. Given two entities (events and/or temporal expressions), the rules evaluate the syntactic expressions in the text between them. In particular prepositions and adverbs are considered for labelling a specific relationship.

[8] http://www.spaziodati.eu/

[9] http://www.timeml.org/publications/timeMLdocs/timeml_1.2.html

The relations considered are those defined in (Allen, 1983): After, During, Start, Meet, etc.. . More details about this service were presented in (Pastore, 2013).

Event-Detector. As presented in section 2, an event is an action that happens over time and involves one or more participants. Starting from this definition, we have an event for each action detected in the text. So we have a service for detecting actions. Such service is defined through a rules-based approach that analyses verbs, nouns, adjectives and adverbs in the text and decides if an expression is an action or not (more details about this service were presented at (Cisternino, 2013)). After the detection of the actions, and knowing the other components (location, temporal expression, agent and physical objects) detected with previous services, the aim is to compose the events associating actions with other components. The composition process is still in the phase of completion and evaluation, but principally, it is based on the analysis of the dependency tree and the relationships between the entities and the actions identified in the text.

Media-Annotation. This service annotates the media using CSWL formalism. At this development stage of HEI, the annotation of media is associated to the whole media (not its parts) or to the caption associated to it (Fig. 6).

4.2 Reasoning services

We can apply various services or axiomatics (formalized in terms of logic programming) to NL texts annotated through CSWL formalism. These services concern temporal reasoning about intervals or events, coherence checking of events, discovery of temporal relationships between events, verification of the connectivity of narrative events, causal reasoning through events. Some details of the following services are described in (Mele et al., 2010; Mele and Sorgente, 2011; Mele and Sorgente, 2013).

Consistency check of events. The consistency check of events regards the consistency of temporal relations in which the events are involved. To implement such a service, the axioms of Russell and Kamp (RK) (Van Lambalgen and Hamm, 2008) have been adopted as input for defining the logical rules (in ASP logic program) that allow the detection of subsequent inconsistency. In particular, a logic ASP program using Clingo system(Gebser et al., 2010), has been developed. The program, given a knowledge base defined through the relations between time points, calculates the stable models, or rather, groups of consistent sets of relations (satisfying the RK axioms). If the program returns more than one stable model, then the relations are inconsistent. Each stable model will contain all consistent relations, if it returns a unique stable model, meaning that the relations between events are consistent.

Discovery of temporal relations among events. The aim of this service is to enrich the set of temporal relations between events. The starting point of this reasoner is to detect a consistent set of events (checked with previous service). Then, once the consistency of temporal relationships has been provided and the only possible stable model (coming from previous service) has been identified, the service identifies new temporal relations between events. For this purpose, a specific set of rules for each temporal relation among events has been defined. Each rule tries to find a particular temporal relationship (before, after, meets, etc.) between two events in accordance with the relations belonging to the stable model returned.

Connectivity check of narrative events. The partial temporal order of events does not ensure that all events belonging to a story are connected. This may be because during the annotation process some relations were not detected and/or because in the text the temporal relations are not present. For this reason, we have implemented a service to evaluate the connectivity of narrative events. The algorithm finds the connected components using a graph representation where the nodes are the events and the edges are the temporal relations between them.

Causal reasoning through events. For causal reasoning service, we have implemented a set of rules (in logic program form) based on the axiomatic defined in (Bochman, 2003)). Through these rules, the system can imply new causal relations, and also provides a consistency check of causal relations. As for the *consistency check of events* service, a logic program through stable model semantics (Gelfond and Lifschitz, 1988) using Clingo system(Gebser et al., 2010), has been implemented.

Persistence fluent reasoner. This set of rules is necessary to evaluate the persistence of all fluent entities present in the CSWL formalism (properties, roles, mental acts, spatial and meronomic relations).

To this end, a service based on Event Calculus has been defined to determine the validity of fluents over time. We have used the *dec.lp* program reported in (Mueller, 2015) and integrated it with others reasoner programs.

Semantic multimedia mashups. A particular aspect concerning this system is the possibility to build *syncretic multimedia artefacts* (Cosenza, 2010; Sorgente et al., 2016a), that semantically synchronise events present in NL texts, extradiegetic voices or captions, with the homologous events present in video annotations. Given a text, this service selects and ranks available media that can be associated to entities annotated in such a text. Both resources, text and media, are annotated by CSWL, so the ranking is based on an index calculated by comparing the CSWL annotation of media with respect to the annotations of the text. It checks if they (media and text) have a common annotation of some entities, that is if the media has annotated entities that are cited in the text. After selecting the media, they are synchronised with synthesised text (through a Text To Speech tool) so that media items are coherently visualised with the relevant time intervals in which a synthetic voice talks about the content represented in the media. An application of such a service has been presented in (Sorgente et al., 2016b).

5 Conclusions

In this work, we have presented the HEI system that helps a user in the annotation of NL text. HEI is composed of various modules, and has as reference ontology CSWL formalism. The HEI architecture is based on services that are implemented as plugins, specialised to execute particular forms of temporal reasoning (named entity recognition, temporal expression recognition, temporal relation extraction, media annotation, consistency check of events, discovery of temporal relations, connectivity check, causal reasoning and persistence fluent reasoning). In HEI there are services for the integration of plugins present in the system. Such a system was used to annotate cultural texts of ’800 exhibit at the Capodimonte Museum and to export them as a Neo4j graph (Fig. 5). Based on these annotations, a dialogue system recovering answers for user’s queries was defined (Origlia et al., 2016; Sorgente et al., 2016b).

References

James F. Allen. 1983. Maintaining knowledge about temporal intervals. Commun. ACM, 26(11):832–843, November.

Alexander Bochman. 2003. On disjunctive causal inference and indeterminism. In IJCAI Workshop: NRAC-03, pages 45–50.

Alessandra Cisternino. 2013. Un sistema ad approccio linguistico per la scoperta di eventi in testi di linguaggio naturale. Tesi di laurea, Università degli Studi di Napoli “PARTHENOPE”.

Giovanna Cosenza. 2010. Semiotica dei nuovi media. Laterza.

Victor de Boer, Johan Oomen, Oana Inel, Lora Aroyo, Elco van Staveren, Werner Helmich, and Dennis de Beurs. 2015. Dive into the event-based browsing of linked historical media. Web Semantics: Science, Services and Agents on the World Wide Web, 35:152 – 158. Semantic Web Challenge 2014.

Martin Gebser, Roland Kaminski, Benjamin Kaufmann, Max Ostrowski, Torsten Schaub, and Sven Thiele. 2010. A users guide to gringo, clasp, clingo, and iclingo.

Michael Gelfond and Vladimir Lifschitz. 1988. The stable model semantics for logic programming. In ICLP/SLP, volume 88, pages 1070–1080.

Lei Hou, Juanzi Li, Zhichun Wang, Jie Tang, Peng Zhang, Ruibing Yang, and Qian Zheng. 2015. Newsminer: Multifaceted news analysis for event search. Knowledge-Based Systems, 76:17 – 29.

Francesco Mele and Antonio Sorgente. 2011. The temporal representation and reasoning of complex events. In CILC, pages 385–399.

Francesco Mele and Antonio Sorgente. 2013. OntotimeFL–a formalism for temporal annotation and reasoning for natural language text. In New Challenges in Distributed Information Filtering and Retrieval, pages 151–170. Springer.

Francesco Mele and Antonio Sorgente. 2015. CSWL - Cultural Stories Web Language. Technical Report 180/15, Institute of Cybernetics “E. Caianiello” of CNR, 06 March.

Francesco Mele, Antonio Sorgente, and Giuseppe Vettigli. 2010. Designing and building multimedia cultural stories using concepts of film theories and logic programming. In AAAI Fall Symposium: Cognitive and Metacognitive Educational Systems.

Erik T Mueller. 2015. Commonsense reasoning: an event calculus based approach. Morgan Kaufmann.

Antonio Origlia, Enrico Leone, Antonio Sorgente, Paolo Vanacore, Maria Parascandolo, Francesco Mele, and Francesco Cutugno. 2016. Designing interactive experiences to explore artwork collections: a multimedia dialogue system supporting visits in museum exhibits. In Proceedings of the 10th AI*IA 2016, volume 1772 of CEUR Workshop Proceedings, pages 26–33. CEUR-WS.org.

Annamaria Pastore. 2013. Un sistema ad approccio linguistico per la scoperta di relazioni temporali in testi di linguaggio naturale. Tesi di laurea, Università degli Studi di Napoli “PARTHENOPE”.

Ansgar Scherp, Thomas Franz, Carsten Saathoff, and Steffen Staab. 2009. F–a model of events based on the foundational ontology dolce+dns ultralight. In Proceedings of the Fifth International Conference on Knowledge Capture, K-CAP ’09, pages 137–144, New York, NY, USA. ACM.

Antonio Sorgente, Antonio Calabrese, Gianluca Coda, Paolo Vanacore, and Francesco Mele. 2016a. Building multimedial dialogues annotating heterogeneous resources. In Luciana Bordoni, Francesco Mele, and Antonio Sorgente, editors, Artificial Intelligence for Cultural Heritage, chapter 3, pages 49–82. Cambridge Scholars Publishing.

Antonio Sorgente, Paolo Vanacore, Antonio Origlia, Enrico Leone, Francesco Cutugno, and Francesco Mele. 2016b. Multimedia responses in natural language dialogues. In Proceedings of AVI*CH 2016, volume 1621 of CEUR Workshop Proceedings, pages 15–18. CEUR-WS.org.

Jannik Strötgen and Michael Gertz. 2015. A baseline temporal tagger for all languages. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 541–547, Lisbon, Portugal. Association for Computational Linguistics.

Willem Robert van Hage, Véronique Malaisé, Roxane Segers, Laura Hollink, and Guus Schreiber. 2011. Design and use of the simple event model (SEM). J. Web Sem., 9(2):128–136.

Michiel Van Lambalgen and Fritz Hamm. 2008. The proper treatment of events, volume 6. John Wiley & Sons.

Paolo Vanacore, Gianluca Coda, Antonio Sorgente, and Francesco Mele. 2016. Creazione di servizi intelligenti da implementare su piattaforme gi esistenti. Istituto di Scienze Applicate e Sistemi Intelligenti “Eduardo Caianiello” (ISASI), CNR. Int. Report 182/1615.

Piek Vossen, Rodrigo Agerri, Itziar Aldabe, Agata Cybulska, Marieke van Erp, Antske Fokkens, Egoitz Laparra, Anne-Lyse Minard, Alessio Palmero Aprosio, German Rigau, Marco Rospocher, and Roxane Segers. 2016. Newsreader: Using knowledge resources in a cross-lingual reading machine to generate more knowledge from massive streams of news. Knowledge-Based Systems, 110:60 – 85.

Xiang-jun Wang, Swathi Mamadgi, Atit Thekdi, Aisling Kelliher, and Hari Sundaram. 2007. Eventory – an event based media repository. In Proceedings of the International Conference on Semantic Computing, ICSC ’07, pages 95–104, Washington, DC, USA. IEEE Computer Society.