UNIVERSITÁ DI PISA
DOTTORATO DI RICERCA IN INGEGNERIA DELL'INFORMAZIONE

AN ONTOLOGY FOR NARRATIVES

DOCTORAL THESIS

Author
Valentina Bartalesi Lenzi

Tutor (s)
Prof. Francesco Marcelloni, Dott. Carlo Meghini

Reviewer (s)
Prof. Franco Niccolucci, Dott. Martin Doerr

The Coordinator of the PhD Program
Prof. Marco Luise

Pisa, December 2016

XXIX

This thesis is dedicated to my beautiful baby boy Riccardo (15/12/2016 - 16/12/2016).

“There’s no certainty, only opportunity.”
V for Vendetta

“Sing, Goddess, Achilles’ rage,
Black and murderous, that cost the Greeks
Incalculable pain, pitched countless souls
Of heroes into Hades’ dark,
And left their bodies to rot as feasts
For dogs and birds, as Zeus’ will was done.
Begin with the clash between Agamemnon –
The Greek warlord – and godlike Achilles.”
Homer, Iliad 1, 1-8
(translated by Stanley Lombardo,
Hackett Publishing Company, Inc.)

Acknowledgements

First of all, I would like to thank my external tutor and CNR chief Dr. Carlo Meghini, for giving me the chance to attend this PhD course but also for theoretically and practically supporting me in writing this thesis and for indicating me the best research ways to follow. Thanks also for revising my thesis giving clever, expert, useful and precious advices.

Thanks to my internal tutor Prof. Francesco Marcelloni for his availability and for supporting me during the entire PhD course. I'm very grateful to my two external evaluators, Dr. Martin Doerr and Prof. Franco Niccolucci for their comments and suggestions that were very useful to improve my thesis and for my future work. Thanks also to Prof. Mirko Tavoni and Dr. Giuseppe Indizio for sharing a bit of their impressive knowledge in the Humanities with me. Among my CNR colleagues, special thanks to Dr. Daniele Metilli who, in the last year, has practically and humanly supported me in my research and to Dr. Caterina Senette for introducing me to the PhD organization and for helping me in any phase of the course of study, with her clever advices and her pure friendship. Thanks also to Filippo, Hanna, Cesare and Anna for being always available and kind with me.

Thanks to all my friends who have ever supported me, but especially in these tragic days.

Finally, thanks to my family. To Gianpaolo, my brilliant physicist, who gave me scientific advices to improve my thesis but especially for supporting and loving me always, but in particular in this moment that is an hell for both of us. I'm proud of you, Gianpaolo, to have been a beautiful dad even if for one day only. To my mother and my father, who practically and spiritually help and support me in this tragic moment of the life of all of us. To my dear grandparents, who love me and cry with me. To Spettro, for giving me always and only pure dog love. To my first baby boy Riccardo, who wrote my PhD thesis with me but left me in the end.

This thesis was supported by the Italian national research project "‘Towards a Digital Dante Encyclopedia"’ (http://perunaenciclopediadantescadigitale.eu/) and by the European Project "‘Parthenos"’ (http://www.parthenos-project.eu/).

Summary

ONe of the main problems of the current Digital Libraries (DLs) is the limitation of the informative services offered to the user who aims at discovering the resources of the DL by queries in natural language. Indeed, all DLs provide simple search functionalities that return a ranked list of their resources. No semantic relation among the returned objects is usually reported that could help the user to obtain a more complete knowledge on the subject of the search. The introduction of the Semantic Web, and in particular of the Linked Data, has the potential of improving the search functionalities of DLs. In this context, the long-term aim of this thesis has been to introduce the narrative as new first-class search functionality. As output of a query, the envisaged new search functionality should not only return a list of objects but it should also present one or more narratives, composed of events that are linked to the objects of the existing libraries (e.g. Europeana) and are endowed with a set of semantic relations connecting these events into a meaningful semantic network. As a necessary step towards this direction, the thesis presents an ontology for representing narratives, along with a tool for the construction of narratives based on the ontology. Moreover, it has used to the tool for evaluating the ontology in the context of an experiment centred on the biography of the Italian poet Dante Alighieri, the major Italian poet of the late Middle Ages. More specifically:

- An overview of the related works developed in the Semantic Web field and in Narratology, and especially in its sub branch named Computational Narratology was reported. The basic principles of Narratology and Computational Narratology have been reviewed along with the study of the Artificial Intelligence literature, especially of the Event Calculus theory, in order to identify the formal components of narratives.

- A conceptualization of narratives has been developed, based on notions derived from narratology and Artificial Intelligence. According to this conceptualization, a narrative consists of a fabula, i.e. the events of a story in chronologically ordered, and several narrations of this fabula (plots), linked to the fabula by an event association relation. A mathematical expression of the conceptualization has been given, in order to provide a characterization of the conceptualization as clear and

as precise as possible, also to be used as a basis for the subsequent development of an ontology of narratives, encoded in OWL. The proposed conceptualization has been validated by expressing it into an existing ontology, the CIDOC CRM, and by endowing it with provenance knowledge, also expressed in a derivation of the CRM, named CRMinf. This expression has been used in the validation experiment, consisting in the modelling a narrative of the biography of Dante Alighieri, provided by a biographer who has scientifically supported this research.

- The population of the created ontology has been performed by means of a semi-automatic approach implemented by a tool for the construction of narratives which obey the ontology. This tool retrieves and assigns URIs to the instances of the classes of the ontology using Wikidata as external resource and also facilitates the construction and contextualization of events, and their linking to form the fabulae of narratives.

- Finally, a qualitative validation of the developed ontology has been carried out. This validation has regarded the evaluation of: (i) the representational adequacy of the ontology by a Dante Alighieri's expert; (ii) the effectiveness of the narrative building tool; (iii) the satisfaction of the users' requirements defined at the beginning of the study. To prove the last point, initial requirements representing pre-requisites of this work have been satisfied by demonstrating that a SPARQL query can be always built to extract the requested information from the knowledge base embodying the narrative.

Sommario

Le biblioteche digitali (BD) sono sistemi informativi che offrono servizi su un vasto insieme di oggetti digitali. Le BD, ufficialmente nate circa venti anni fa negli Stati Uniti d'America nell'ambito della Digital Library Initiative, negli ultimi anni hanno avuto una notevole diffusione grazie agli sviluppi in campi tecnologici rilevanti come le tecnologie della comunicazione, i sistemi multimediali, le ontologie e il web semantico. Grazie a questi progressi oggi le BD mettono a disposizione dei loro utenti servizi che coprono un vasto spettro di tipologie di oggetti digitali, dagli ipermedia ai modelli 3D.

Uno dei principali servizi informativi delle attuali BD è l'evoluzione di un servizio presente nelle biblioteche tradizionali: supportare gli utenti nel discovering di oggetti, digitali o meno, che soddisfino il loro bisogno informativo, tipicamente espresso come una query in linguaggio naturale, consistente in poche parole chiave. Il discovering funziona ragionevolmente bene sul Web, che può essere visto come un'immensa biblioteca digitale nella quale gli oggetti sono le pagine iper-testuali ricche di contenuto testuale, visuale e grafico, e collegate le une alle altre attraverso link non tipizzati. La situazione non è altrettanto favorevole all'utente nelle BD propriamente dette, come Europeana, la biblioteca digitale europea, che raccoglie oltre trenta milioni di record di metadati da oltre due mila fra le maggiori istituzioni culturali europee (www.europeana.eu). Tali BD rispondono a una Web-like query proponendo all'utente una lista di descrittori di oggetti, offerti in accesso dalla biblioteca, senza esplicitare alcun tipo di relazione semantica tra i risultati.

Una soluzione a questo problema potrebbe essere costituita dall'introduzione di una nuova funzionalità di ricerca per le BD che le metta in condizione di fornire all'utente, in risposta ad un'interrogazione, non più una lista di oggetti, ma una narrazione formale, composta sulla base degli elementi contenuti nella biblioteca digitale, sul tema dell'interrogazione. Per narrazione formale si intende una rete semantica che rappresenta gli eventi che compongono la storia narrata e i loro sotto-eventi, e che collega tali eventi tra di loro e con gli oggetti della biblioteca digitale tramite apposite ed esplicite relazioni semantiche. Il concetto di evento è un concetto chiave nella letteratura relativa alla narratologia e alla narratologia computazionale, e indica un accadimento o feno-

meno fisico situato nello spazio-tempo. L'uso delle narrazioni formali nelle BD può, inoltre, essere utile per due aspetti non strettamente collegati con la discovery: (i) per rappresentare la provenance della conoscenza e degli oggetti di una BD, ovvero la descrizione dell'origine e del processo tramite il quale conoscenza e oggetti arrivano a far parte di una BD; la rappresentazione della provenance è cruciale per la determinazione del livello di trust che un utente ha dei contenuti della BD, nonché la conservazione di tali contenuti; e (ii) per arricchire il contenuto delle BD con eventi che connettono gli oggetti in unità ricche di significato in modo da facilitare l'interpretazione del contenuto della BD.
Come passo necessario verso l'introduzione delle narrazioni nelle BD, questa tesi presenta un'ontologia per la rappresentazione formale di narrazioni ed un tool semi-automatico che sulla base dell'ontologia permette di costruire e visualizzare narrazioni ben formate.
Il tool è stato, inoltre, utilizzato per valutare l'ontologia nel contesto di un esperimento di creazione di una narrazione incentrata sulla biografia di Dante Alighieri, il più importante poeta italiano del Medioevo.

List of publications

International Journals

1. Bartalesi, V., & Meghini, C. Using an ontology for representing the knowledge on literary texts: The Dante Alighieri case study. Semantic Web Journal, vol. Preprint, no. Preprint, pp. 1-10, DOI: 10.3233/SW-150198, IOS Press, 2015.
2. Bartalesi, V., Meghini, C., Andriani, P., & Tavoni, M. (2015). Towards a Semantic Network of Dante's Works and Their Contextual Knowledge. Digital Scholarship in the Humanities, 30(suppl 1), i28–i35, DOI: http://dx.doi.org/10.1093/llc/fqv044, Oxford University Press, 1 October 2015.

International Conferences

1. Bartalesi, V., Meghini, C., & Metilli, D. (2016, July). Steps Towards a Formal Ontology of Narratives Based on Narratology. In 7th Workshop on Computational Models of Narrative (CMN 2016) (pp. 41–50). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl-Germany.
2. Bartalesi, V., Meghini, C., Metilli, D., & Andriani, P. (2016, July). Usability Evaluation of the Digital Library DanteSources. In International Conference on Human-Computer Interaction 2016 (pp. 191–203). Springer International Publishing.
3. Bartalesi, V., & Leporini, B. (2015, August). An Enriched ePub eBook for Screen Reader Users. In International Conference on Universal Access in Human-Computer Interaction (pp. 375–386). Springer International Publishing.
4. Tavoni, M., Andriani, P., Bartalesi, V., Locuratolo, E., Meghini, C., & Versienti, L. (2014). Towards a semantic network of Dante's works and their contextual knowledge. In Proceedings of The Digital Humanities 2014 conference. Lausanne, Switzerland. July (pp. 7–12).
5. Meghini, C., & Bartalesi, V. (2014, June). Steps towards Enhancing the User Experience in Accessing Digital Libraries. In International Conference on Human

Interface and the Management of Information (pp. 555–566). Springer International Publishing.

6. Bartalesi, V., Buzzi, M. C., Buzzi, M., Leporini, B., & Senette, C. (2014, June). An Analytic Tool for Assessing Learning in Children with Autism. In International Conference on Universal Access in Human-Computer Interaction (pp. 209–220). Springer International Publishing.

7. Bartalesi, V., Locuratolo, E., Versienti, L., & Meghini, C. (2013, September). A preliminary study on the semantic representation of the notes to Dante Alighieri's Convivio. In Proceedings of the 1st International Workshop on Collaborative Annotations in Shared Environment: metadata, vocabularies and techniques in the Digital Humanities (p. 4). ACM.

Others

1. Bartalesi, V., Meghini, C. (2016). Formal Components of Narratives. Post Proceedings of the Italian Research Conference on Digital Libraries (IRCDL) 2016. Communications in Computer and Information Science (CCIS), Springer International Publishing (under publication).

2. Meghini, C., Bartalesi, V., Metilli, D. (2016). Steps Towards Accessing Digital Libraries Using Narratives. Proceedings of The 10th Workshop on Artificial Intelligence for Cultural Heritage (under publication).

3. Bartalesi, V., Meghini, C., Metilli, D., Andriani, P., & Tavoni, M. (2016) DanteSources: a Digital Library for Studying Dante Alighieri's Primary Sources. Proceedings of AIUCD Annual Conference 2015 (under publication).

Contents

CHAPTER 1

Introduction

1.1 Motivation

Digital Libraries (DLs) are information systems that offer information services over large sets of digital objects [93]. DLs were officially born about two decades ago in the Digital Library Initiative[1]. In two successive rounds, this initiative funded several projects and brought DLs in the focus of research. Much progress has been achieved since then, mainly due to major breakthroughs in relevant technological fields, such as communications, information retrieval, ontologies and multimedia. Thanks to these technological progresses, today's DLs have a wide penetration and cover a wide spectrum of digital object types, ranging from hypermedia to 3D models. However, the basic information service of a contemporary DL is essentially the same as that of a traditional library: to support users in *discovering* the digital objects that satisfy an information need, typically expressed as a query consisting of a short list of terms. This kind of discovery service works reasonably well on the web, which may be viewed as a very large DL whose objects are pages rich in textual content and interlinked between each other. On the contrary, the traditional discovery functionalities of DLs, such as Europeana[2], respond to a web-like query with a ranked list of digital objects based only on metadata descriptors that are semantically poor. For example, consider as user a young woman wishing to know more about Dante Alighieri, the major Italian poet of the late Middle Ages. She may ask "Dante Alighieri" to her favourite web search engine and most likely she would get a list of ranked documents with the Wikipedia page about Dante within the top 5 results. Not willing to spend her time reading (which she can do in her favourite traditional reference library), the user tries other web sites, where she

[1] http://www.dlib.org/dlib/july98/07griffin.html

[2] Europeana is the largest DLs in the cultural heritage sector, offering a unique access point to about forty million objects from more than two thousand European institutions. See `http://www.europeana.eu/portal/`

hopes to find something quicker and more exciting to consume than the typical textbook narrative. At some point, she lands on the web page of a DL, say Europeana, where she tries again her query. To her disappointment, the result is a long list of disparate objects, each offering a glimpse of Dante's life and works, but altogether incapable of providing an idea of who Dante was and what he did. The user tries other digital libraries, getting the same experience, that is being presented with list of objects that the library has relative to Dante Alighieri. This generalized behaviour is a consequence of seeing a digital library as a traditional library endowed with digital resources managed by software. However, the behaviour is strictly in contrast with the very idea at the core of a digital library. The final report of an NSF sponsored Santa Fe planning workshop on distributed knowledge work environments[3] held in March 1997, defined a "digital library" as follows: "...the concept of a digital library is not merely equivalent to a digitized collection with information management tools. It is rather an environment to bring together collections, services, and people in support of the full life cycle of creation, dissemination, use, and preservation of data, information, and knowledge.". After almost twenty years of research and development, the above concept is still out of reach of today's digital libraries. This thesis is based on the idea that in order to realize the above vision, DLs should offer something more than a ranked list of objects to information seekers. In particular, DLs should be able to provide *narratives* to their users. Narrative is a well-researched concept in several fields, ranging from literary studies to cognitive science. Giving an account of the concept is beyond the scope of this thesis. As a matter of fact, "narrative can be viewed under several profiles - as a cognitive structure or way of making sense of experience, as a type of text, and as a resource for communicative interaction" [62]. For the purposes of this research, a narrative was intended as a network of "temporally indexed representations of events" [95], that is events associated to time structures and related to one another and to the DL resources through semantic links. Indeed, the goal of this study is to promote narratives as first-class objects for DLs. Thanks to this this new feature, DLs will be able to semantically connect objects to events, and through events to other objects, resulting in a much richer information structure. Such information structure will allow DLs to provide more sophisticated information services to their users, making an important step towards the creation of the "environment" that the above definition alludes to, and as such going beyond the current state.

1.2 Research Proposal

Based on the considerations developed in the previous Section, this thesis presents the study and the process to introduce narratives in digital libraries as a first-class new citizens, structured according an ontology, expressed using the Semantic Web languages. In proposing this approach, the thesis relies on a few fundamental assumptions. First, it is assumed that the extensive research carried out on narratives in the last decades has been successful in identifying a set of common structures that characterize narratives. Moreover it is assumed that such characteristics are amenable to formal treatment, and in particular can be adequately formalized in an ontology. Finally, it is assumed that the developed ontology can be experimentally validated by re-using existing ontologies.

[3] http://www.si.umich.edu/SantaFe/

As for the first assumption, it suffices to say that narratology is a well consolidated discipline since at least two decades, offering well-understood notions for the conceptualization. Chapter 2 will elaborate on this point, by providing an extensive review of the narratology literature, being it granted that the vastity of such literature prevents an exhaustive account. The same Chapter will also show that the emerging characteristics of narratives can indeed be captured in symbolic terms - more specifically, propositional terms much in the same way a scientific vocabulary is captured by logical axioms. Hence an ontology is the natural way of conveying such a characterization. As for the last assumption, the problem of selecting a knowledge representation framework that would facilitate in the experimental validation of the conceptualization and, at the same time, would be as credible as possible to the potential adopters of the developed technology was faced. This problem has two aspects: from the one hand, it requires the selection of a syntax for expressing knowledge, with an associated machine-readable notation, the so-called logical symbols of the sought language; from the other, it requires the selection of a suitable set of predicates corresponding to the notions that are relevant in narratology, the so-called non-logical symbols. As for the former aspect, the choice has been the Semantic Web family of languages and related technologies. This choice is motivated by the large and increasing adoption of semantic web technologies in the context of DLs and beyond. These languages are W3C Recommendations since more than a decade, and as such propose themselves as de facto standards. As for the latter aspect, the CIDOC CRM ontology (or simply CRM) [4] was chosen. The CRM is an ISO (hence de iure) standard, and one of the most comprehensive ontologies to describe the cultural-historical domain, hence catering in a natural way to the needs of DLs. As it will be shown, the CRM is adequate also to explore the questions that arise from the formalization of the components of narratives, unlike its more library-oriented competitors such as the Dublin Core.

In accordance with the programme, the main phases of work have been the following:

1. development of a conceptualization of narratives based on notions found in narratology and in the Artificial Intelligence (AI) literature;

2. mathematical specification of the conceptualization, as a necessary pre-requisite to the

3. expression of the specification in a standard ontology, the CRM ontology, to the end of carrying out a

4. validation of the developed ontology on a realistic case study: the representation of the biography of Dante Alighieri; the implementation of this case study in turn required the

5. design and implementation of a semi-automatic tool to create narratives from texts, and to visualize them in a user-friendly way.

The main outcome of the thesis and its main contribution to the current state of research on DLs are:

[4] http://cidoc-crm.org/

- a validated conceptualization to represent narratives, and its experimental expression in the CIDOC CRM, and
- a semi-automatic software tool to create and visualize narratives, which can been easily integrated in a DL. This will be mainly achieved using semantic Web technologies.

1.3 Outline

The thesis is structured as follows: Chapter 2 gives a definition of "ontology" and introduces the ontology languages. It also introduces the basics of the Narratology and Computational Narratology theories along with an overview of narratives in Artificial Intelligence. Chapter 3 present the main problem that is investigated in this thesis, i.e. the fact that the free-text queries on digital libraries usually return a ranked list of the information objects but no semantic relation among them is reported that can help the user in obtaining a more complete knowledge on the subject of the research, the methodological approach followed in this study and an analysis of the users' requirements. Chapter 4 presents the developed ontology for representing narratives. In order to derive the conceptualization of narratives, the Event Calculus is first analysed. Further, two standard ontologies are analysed in order to understand whether their vocabularies are rich enough to capture the logic definitions of the components of narratives (Section 5.1), i.e. CIDOC CRM and DOLCE+. Then, a conceptualization is presented, which introduces a formal computable representation of narratives in an informal way, based on the narratology background. Finally, a mathematical specification of the conceptualization is provided. In Chapter 5 an experimental implementation of the mathematical specification using CRM is reported and the ontology is used to represent a case study, i.e. the narrative of the biography of Dante Alighieri, the major Italian poet of the late Middle Ages. Chapter 6 describes a semi-automatic tool to populate the created ontology. This tool retrieves and assigns URIs to the instances of the classes of the ontology, using the Wikidata knowledge base as resource. Further, it facilitates the construction and contextualization of events, and of their links to form the narratives. In Chapter 7 an evaluation of the ontology and the tool developed is presented. In particular, the following aspects are evaluated: (i) the representational adequacy of the ontology by a historian who is an expert of Dante's life; (ii) the effectiveness of the tool to build narratives; (iii) the satisfaction of the users' requirements identified at the beginning of this study. Finally, Chapter 8 reports conclusive remarks and future work.

CHAPTER 2

An Overview of Computational Narratology

In this Chapter a review of some fundamental notions from the theory of narratology and from the Event Calculus was given, along with some definitions from the general area of Artificial Intelligence.

2.1 Narratology

In literary theory, narratology is a discipline devoted to the study of the narrative structure and the logic, principles, and practices of its representation [62, 96]. The earliest antecedent to modern narratology can be found in the classical Aristotle's theory of aesthetics. Indeed in *Poetics*, Aristotle defines a narrative as the imitation of real actions (*praxis*) that forms an argument (*logos*) whose fundamental units, or events, can be arranged in a plot (*mythos*) [7].

For the Russian formalism the narratology is based on the idea of a common literary language, or a universal pattern of codes that operates within the content of a work. A narrative can thus be conveyed through several different means of communication and a wide range of media, including speech, writing, gestures, music, etc. In particular, Vladimir Propp's *Morphology of the Folktale* (1928) [114] proposed a model to represent folktales as combinations of basic building blocks, including 31 “”narrative functions” and seven roles, or “spheres of action”, of the characters.

The theory of narratology was further developed by mid-20th Century structuralism. Claude Lévi-Strauss, in *Structural Anthropology* (1958) outlined a grammar of mythology; in *Structural Semantics* (1966) [54] A.J. Greimas proposed a system of six basic structural elements of narratives called *actants*; Tzvetan Todorov was the first to coin the French term *narratologie* [127]. Later on, Gérard Genette [50] codified a system of analysis that studied both the narration and the act of narrating, considering them separately from the story and content of the text.

Since 1980, post-structuralist perspectives of narratology have been developed [46]. In particular, Cognitive Narratology [61] considers narratology a psychological phenomenon, and proposes a study of narrative aspects from a cognitive perspective. Empirical results from cognitive psychology highlight that most common-sense concepts cannot be characterised in terms of necessary/sufficient conditions. A monotonic description logic can capture the aspects of conceptual knowledge, but are insufficient to represent prototypical knowledge. However, a general description logic to represent concepts in prototypical terms does not exist yet [49, 76].

2.1.1 Fabula and Syuzhet

The Russian formalism distinguishes between *fabula*, defined as a series of events taking place at a certain time at a specific location, and *syuzhet* (or plot), which is the particular way the story is narrated. At opposed to the strictly chronological order of the fabula, the order of the syuzhet corresponds to the way the events are presented in the narrative by the author [114] [121]. A similar distinction is drawn in structuralism by Chatman [23], who identifies the opposing concepts of *story*, i.e. the content that is transmitted, and *discourse*, i.e. the particular organization of that content.

Currently, there is no universally accepted definition of narrative structure. For instance, Crawford [28] posits that a narrative is a high-level structure based on causality, but it is not based on temporal or spatial relations. Genette [50] identifies five concepts that characterize the syntax of narratives: order, frequency, duration, voice and mood. In addition to the fabula and the syuzhet, Bal [9] defines a third level, called *presentation*, that constitutes the concrete representation of the content that is conveyed to the audience (e.g., the text in a novel).

2.1.2 Characters and Plots

In narratology, two entities are particularly important: characters and plots.
Characters are a fundamental constituent in a story. Aristotle considered that the most important element in the action [7], but characters appear in every type of tale. McKee [117] claims that it is not possible talking about the plot without characters and vice versa. Chatman [24] distinguishes the elements of a story in characters and elements in the scenario. Characters are usually humans or humanoid beings while the elements in the scenario are places and objects.
The terms plot refers to the events that build a story. In particular, the term refers to the relation that links an event to another in a pattern.

2.1.3 Fictional and Non-fictional Narratives

A dichotomy proper to narratology regards the distinction between fictional and non-fictional narratives [65]:

- A fictional narrative has mainly an entertainment value, and represents an imaginary narrator who tells a story happening in an imaginary world.

- A non-fictional or factual narrative presents, on the contrary, a real-life narrator of a real-life story. Factual narratives (like news stories) represent evidences of what happened in the real world.

This dichotomy is taken into account in [138], where the author presents a system and a formal language (NKRL) for narratives representation. This work focuses on the representation of non-fictional narratives only and the proposed language deals mainly with this kind of narratives. On the contrary, the semantic model developed in this thesis represents both fictional and non-fictional narratives, because the objects of the knowledge representation are general, e.g. essays, news stories, fairy tales etc.

2.2 Computational Narratology

Computational Narratology studies narratives from a computational perspective. In particular, it focuses on "the algorithmic processes involved in creating and interpreting narratives, modelling narrative structure in terms of formal computable representations" [84].
The term "Computational Narratology" (CN) can assume different meanings according to the research context. In particular, in the context of Humanities, computational narratology is defined as a methodological instrument for constructing narratological theories, extending narratological models to larger bodies of text, providing precise and consistent explication of concepts [95]. From a cognitive computing point of view, this terms refers to narrative texts, computer games, and more in general, software developed using semiotic, sociolinguistic and cognitive linguistic theories [60]. From the point of view of Artificial Intelligence (AI) [22], CN refers to story generation systems, i.e. computer applications that create a written, spoken, or visual presentation of a story. In the context of this thesis, the AI domain is the more interesting among the above views, thus some of its aspects are in investigated in this Chapter. Two disciplines are associated to CN in the domain of Artificial Intelligence: *eChronicles* and *storytelling*.

An eChronicle system can be defined as a way of recording, organizing and accessing streams of multimedia events captured by individuals, groups, or organizations making use of video, audio and other sensors. In this perspective a narrative can be represented by meeting minutes, conference records, surveillance videos, visitor logs, sales activities, etc. This material is aggregated into coherent episodes of narratives in order to allow access to this material at different levels of granularity [59, 134]. Storytelling [124] generally refers to the study of different ways of conveying stories and events in words, images and sounds, in order to entertain, teach, explain, etc. In particular, *digital storytelling* refers to the ways of introducing characters and emotions in interactive entertainment, e.g. video games, multi-playing, online games, interactive TV, virtual reality, etc. [97]. Digital storytelling is related to another computer-based variant of narratology named *narrative intelligence*, a specific sub-domain of AI that investigates topics that fall in the intersection between AI, media studies and human computer interaction design. Examples are narrative interfaces, history databases management systems, artificial agents with narratively structured behavior, systems for the generation and/or understanding of histories and narratives, etc. [87].

2.2.1 Narratives in Artificial Intelligence

Artificial Intelligence (AI) focuses on the study of the intelligence exhibited by machines or software and reports about how this intelligence can be created. AI studies involve intelligent "agents" the study of the behaviour of an agent that answers to environmental changes and performs actions [118]. Artificial agents build their own representation of the world and use an inference process to create new representations and consequently defining the resulting actions. Knowledge and reasoning are key concepts for agents since they allow (i) inferring the consequences of actions and (ii) combining general knowledge with singular environment inputs to infer some hidden aspects before selecting an action. For this reason, knowledge is crucial in partially observable and dynamic environments, and when agents need to be given new tasks. The core component of artificial agents is a knowledge base (KB). Informally speaking, a KB is a set of sentences. The sentences are expressed in a knowledge representation language. The KB supports "tell" and "ask" tasks. The "tell" operation adds a new sentence to the KB. The "ask" operation reports if a sentence is a consequence of the content of the KB. When a knowledge-based agent either receives a percept or performs an action, it "tells" this to the KB. An agent's designer can populate the KB with initial knowledge from which agents can infer new knowledge. The sentences of the KB are expressed using a syntax defined by a representation in a logic language. The logic defines also the semantics of the sentences. In logics, the semantics defines the truth of each sentence with respect to possible worlds. In this context, the logical reasoning can be defined as the logical entailment between sentences. The entailment is a logical consequence, and then the idea is that a sentence logically follows from another one. Many of the problems machines are expected to solve, require extensive knowledge of the "world". To manage this knowledge, AI needs to represent a number of elements, e.g. objects, properties, categories and relations between objects; situations, events, states and time; causes and effects [113]; knowledge about knowledge (what we know about what other people know) [118].

Knowledge engineering aims at integrating knowledge bases into computer systems to find a solution for complex problems usually solved by human experts [42]. This discipline consists of six steps that transfer human knowledge into knowledge for computer systems. The resulting process is named ontological engineering:

- Identification of a task: The knowledge engineer defines the task that the knowledge base will support and how knowledge should be represented in order to connect questions with answers.

- Knowledge acquisition: The knowledge engineer chooses the relevant knowledge for the task. At this stage, the knowledge is not formally represented yet.

- Definition of a vocabulary: In this phase the engineer translates the concepts at the discourse level into names at the logic level. The result is a vocabulary, named "ontology of the domain". An ontology is a formal naming and definition of the classes, properties, and interrelations that actually or fundamentally exist for a particular domain of discourse. The most general ontologies are called "upper ontologies". These contain general concepts at the top of the graph and specific concept at lower levels. Upper ontologies have general-purpose and provide the base for the full range of human knowledge.

- Encoding of knowledge about the domain: The knowledge engineer writes down the formal axioms for the ontology terms.
- Encoding a description of the specific problem instance: This step involves writing atomic sentences about instances of concepts that are part of the ontology.
- Queries and answers: An inference engine operates on the axioms to derive the facts that agents or KB clients are interested in.

2.2.2 Story Generation Systems

Storytelling systems aim at reproducing a human-like narrative behaviour or at creating interfaces or game environments using narratives as interactive methods. Early storytelling systems were TALE-SPIN [92], UNIVERSE [75], GESTER [109] and JOSEPH [72]. These system were able to change the story grammars to create new stories. Other storytelling systems are MINSTREL [128], MEXICA [110] and BRUTUS [19]. These are hybrid systems that implement a computer model of creativity in writing. Recently, ontologies have been used to generate narratives. For example, MAKEBELIEVE [78] uses common-sense knowledge, selected from the ontology of the Open Mind Common-sense Knowledge Base [122], to generate short stories from an initial one given by a user. ProtoPropp [51] uses an ontology of explicitly relevant knowledge and the Case-Based Reasoning method over a defined set of tales. In FABULIST [116] the user supplies a description of an initial state of the world and a specific goal, and the system identifies the best sequence of actions to reach the goal.

The concept of *event* is a core element of narratology theory and of narratives. People conventionally refer to an *event* as an occurrence taking place at a certain time in a specific location. Various models have been developed for representing events on the Semantic Web, e.g. Event Ontology [115], Linking Open Descriptions of Events (LODE) [120], the F-Model [119]. More general models for semantic data organization are the CIDOC CRM [36], the ABC ontology [71] and the Europeana Data Model [35].

Narratives have been recently proposed to enhance the information contents and functionality of Digital Libraries, with special emphasis on information discovery and exploration. For example, in the CultureSampo project [63] an application to explore Finnish cultural heritage contents on the Web, based on Semantic Web technologies was developed. This system uses an event-based model and generates links among events and digital object. However, it does not allow visualizing events and their related digital objects as a semantic network endowed with semantic relations that connect events and objects. Another example is Bletchley Park Text [103], a semantic application helping users to explore collections of museums. Visitors express their interests on specific topics using SMSs that contain keywords. The semantic description of the resources is used to organize a collection into a personalized web site based on the chosen topics. The system relies on an ontology of stories, taken from the Story Fountain project [104]. The stories represented in the system are exploited to create relations between the entities contained in the online collections, allowing a user to query the system for a semantic path between entities. In the PATHS project [43], a system that acts as an interactive personalized tour guide through existing digital library collections is proposed. In this system, the events are linked by inherence relations. Similarly to the approach of PATHS, the CULTURA project [1] proposes a tool to enrich cultural heritage collections with guided paths in the form of short lessons named *narratives*. The Storyspace system [137] allows describing stories based on events that span museum objects. The system focuses on the creation of curatorial narratives from an exhibition. Each digital object has a linked creation event in its associated heritage object story. Kilfeather and McAuley (2003) [68] describe some tools which would facilitate the development of a meaningful story or narrative structure from existing or new contents. The aim is to allow authors to establish semantic relations between different contents and select and merge them. The ontology was based on existing taxonomies and thesauri related to Irish archaeology. The OntoMedia ontology [74] allows representing heterogeneous media through the description of the semantic content of the media (e.g. literary texts, TV program). The representation may be limited to the description of some or all the elements contained in the source or may include information regarding the narrative relationships that these elements have both to the media and to each other. Another example is the Cadmos project [80] that adopts a computer-supported tool to semantically annotate narrative media objects (video, text, audio, etc.) and integrates a large common-sense ontological knowledge.

2.2.3 Ontology Learning Approach

Ontology learning (OL) is a method to create ontologies in an automatic or semi-automatic way. Manually building an ontology can be difficult and time consuming. For this reason, developing an automatic process for ontologies construction would be very useful to ontology engineers. The creation of an ontology in a certain domain starts from the extraction of domain terms and relationships between domain concepts, based on a corpus of natural language text. At the same time, the terms and the relationships are encoded with an ontology language. There are many semi-automatic approaches to construct ontologies starting from (structured or unstructured) texts or Web pages. These approaches use basic techniques developed in data mining, natural language processing, computational linguistics or machine learning. Currently, several OL approaches automatically build ontologies using different sources like WordNet or the World Wide Web as knowledge sources, in addition to natural language corpora. Generally speaking, the quality of the ontologies developed using semi-automatic methods is still worse than the quality of expert's developed ontologies. Indeed, without a pre-processed input and a revision by an ontology engineer, the produced ontologies are not usable in many cases. This concept is clear reading the brief overview of some of the main OL systems reported below.

Ontology-like structured knowledge from scientific literature

This method developed by Blaschke and Valencia (2002) [15] is based on a domain-specific corpus of molecular biology literature. The aim of this approach is to introduce a methodology to build an ontology from literature studies to represent the relations between the genes of an organism. In order to identify genes references in documents, the exact names of the genes along with two orthographic variances are extracted. Using basic statistical methods, lists of keywords are extracted to describe their contents. Documents are grouped using clustering analysis and similarity between these groups are calculated. A hand-made ontology is built after comparing the ontology extracted from scientific literature and the Gene Ontology [8], a hand-made ontology that represents the classification of a large number of protein functions.

Automatic ontology extraction from unstructured texts

In Ahmad and Gillam (2005) [2], a statistical approach is used to identify domain-specific keywords for nuclear physics, by calculating the difference between the relative frequencies of words in domain specific texts and their relative frequencies in a general corpus of language. Collocation analysis [112] is used to identify words that occur frequently within the same context and that are used to communicate specific meanings. This analysis is useful for discovering complex concepts within domain specific texts. The identified patterns can be used to produce collocational networks [136]. Is-a relationships are defined between words and their collocations and hierarchies are produced from these networks.

ONTOGEN

ONTOGEN [47] is a tool to build *topic* ontologies. A topic ontology is a set of topics linked by relations. Each topic is associated with a set of related documents. The ONTOGEN tool allows a user to construct a topic ontology from a corpus of documents in a semi-automatic manner, possibly suggesting new topics and allowing to visualize the ontology. Two different approaches to discover topics within the corpora are used:

1. Latent Semantic Indexing (LSI) that calculates if words related to the same topic co-occur together more often than words related to the different topics, using mathematical transformations. The result of LSI applied to the documents are fuzzy clusters of words each describing one topic.

2. The K-Means clustering algorithm that partitions the corpus into K clusters so that two documents within the same cluster are more closely related than two documents belonging to two different clusters.

Other interesting features are customization and collaboration features, i.e. users can choose to work on ontologies in cooperation with other users. Overall, the produced ontologies are simple and require user intervention to be further refined.

Text-To-Onto

A fairly elaborate approach is the Text-To-Onto system developed at the University of Karlsruhe by Maedche et al. [81–83]. Text-To-Onto has been further developed by Cimiano and Volker [111] and Cimiano [26], where it has been renamed as Text2Onto. The input of the system is constituted by structured or unstructured data like databases, texts, web pages etc. These sources are pre-processed and then a shallow text parser is used. The parsed documents are used by two groups of algorithms. The first group is based on frequencies of words in the text and addresses the identification of concepts: a concept hierarchy is created using a clustering algorithm and a rule-based association algorithm detects non taxonomical relations. Then, the names of these relations are manually assigned. The second group of algorithms uses pre-defined lexicon-syntactic patterns to extract both taxonomical and non-taxonomical relations. The relations are presented to the user who can accept or discard them. This system uses an ontology engineering environment (KAON[1]) that lets the user manually edit the produced ontology.

OntoLearn
The OntoLearn system [105], uses WordNet [98] and SemCor [99] as knowledge sources and a knowledge base containing annotated sentences to interpret terms. A shallow parsing is used to extract possible relevant terminology from a text corpus, both in the form of single words and complex phrases. The frequency of the keywords in the domain corpus are measured with respect to a corpus covering several domains. A second measure is the domain consensus, which measures how many different documents the term or phrase occurs in. For every word contained in the trees built by the parser, a semantic net is created based on the sense given by WordNet and SemCor. In order to connect all the semantic nets, taxonomic relations are inferred using WordNet. OntoLearn is oriented to language processing applications and one weak point is that several resources need to be specified in advance, e.g. the relation types available for the non-taxonomic relations.

TextStorm
This system developed by [107] uses a parser over a corpus of texts. The system tags the text using WordNet as reference resource and then retrieves predicates. Predicates represent relations between the terms extracted from the sentences. The synonyms sets (synsets) of WordNet are used to avoid extracting the same concept several times, when denoted by a different term. The output of TextStorm is not an ontology, but rather a list of predicates. Another tool (Clouds) supports the actual building of the "concept map", i.e. the ontology, in a collaborative and interactive way. It uses a machine learning algorithm to pose questions to the user and to draw conclusions depending on the answer.

ASIUM
This is a system that uses linguistic patterns and machine learning in order to create taxonomies [40, 41] rather than ontologies. A text corpus is used as input to produce syntactic frames, i.e. instances of common sentence patterns consisting of combinations of verbs, prepositions and keywords. A machine learning algorithm is used that relies on the assumption that "words occurring together after the same preposition and with the same verbs represent the same concept" [41]. A frequency measure is used to identify which concepts are more important than others.

LExO
LExO [132, 133] is a system that transforms sentences written in natural language into OWL-DL expressions. The input needed for generating DL expressions is very specific, i.e. it is made up of sentences that are written in the form of dictionary definitions. The sentences are parsed by a dependency parser, and the parse tree is translated into an XML-based format, whereby a set of rules can be applied. Hypotheses for disjointness axioms can be generated by applying a few general heuristics. The implementation of LExO relies on KAON2[2] an ontology management infrastructure for OWL DL, and on the Minipar parser[3]. Given a natural language definition of a class,

[1] http://www.aifb.kit.edu/web/KAON/en
[2] http://kaon2.semanticweb.org/
[3] https://gate.ac.uk/releases/gate-7.0-build4195-ALL/doc/tao/splitch17.html

LExO analyses the syntactic structure of the input sentence and generates a dependency tree. This tree is then transformed into a set of OWL axioms by means of manually engineered transformation rules.

Boxer
Boxer [17] is a system for semantic analysis of natural language. Based on a grammar and background information, Boxer produces a representation of the discourse of the input sentences, by translating natural language sentences into logical formulae. The system is highly dependent on the input grammar and the predefined rules that were used to tag the input texts.

OntoLT
OntoLT [20] is a tool that helps users to extract ontological components from preprocessed texts. It is a plug-in of Protege[4] that allows defining patterns to map elements of tagged texts onto ontology concepts.

Summary

Most of the reported approaches use supervised NLP techniques that rely on computer science systems trained on specific domains of application. For example, Parsers, Part-of-Speech taggers, Chunkers [67] are commonly used techniques that support OLs. In many cases, OLs approaches automatically extract ontology terms from the texts according to their frequency of occurrence and of their semantic values [2, 15]. Several systems using parsers are language dependent and “style dependent”: they cannot be applied neither to texts written in other languages than the one on which they have been trained, nor to texts written in other styles (e.g. medieval, poetic texts). Apart from approaches trained on specific domains, some OLs use wide-domain texts as reference to detect terms in specific-domain texts by evaluating relative similarities of terms usages [73, 107]. Nevertheless, they also suffer from language and style dependency. One practical aim of this thesis is to build a narrative system that allows users to create their own narratives, based on text analysis that could use different styles at the same time, e.g. a set of biographies of Dante Alighieri that involved medieval texts as well as modern texts. Furthermore, the system should work also if the text were written in another language than Italian and should be applied to a wide range of domains. In other words, the proposed system addresses the building of narratives that are (i) domain independent, (ii) style independent and (iii) language independent. For this reasons an OL approach was not used in the ontology implementation phase. Rather, an OL approach is proposed as a refinement tool after the creation of the ontology.

2.2.4 Computational Models of Narrative

Computational models of narrative are semantic model developed in order to represent the different aspects that characterize the narratives. The workshop on Computational Models of Narrative investigates this field of research. This is an international initiative started in 2010 with three main objectives: (1) understanding the scope and the

[4] http://protege.stanford.edu/

dimensions of narrative models, identifying gaps and next steps; (2) evaluating the state-of-the-art, and (3) building a community focused on computational narrative [45]. There were three common denominators among the representations presented at the workshop: (1) narratives have to do with sequences of events; (2) narratives have hierarchical structure; and (3) they are grounded in a common-sense knowledge of the world. In this context many papers introducing the main aspects of computational models of narrative were published. In the following, a brief overview is reported about the complexity of narratives modelling and of several different aspects of narratives that a knowledge engineer has to take into account when developing computational models of narratives. [13] is focuses on the belief in a story, intended as coherent sequence of events. The belief is determined by looking at how well the story fits both the available evidence and the common-sense knowledge. This is an interesting premise because representing the process that allows creating a knowledge is crucial in narratives modelling. Another important concept introduced in [13] is the notion of *story schemes*, i.e. general patterns of events that can serve as a background to particular stories. Story schemes range from abstract to specific, for example a scheme like "beginning-middle-end" is a very abstract scheme for stories. In [138], the author presents a system to automatically manage narratives based on a language for narratives representation called NKRL. This language is built upon a simple grammar, based on three sets of symbols: (i) conceptual predicates, which define the types of events, (ii) the roles that establish the structure of an event based of the involved entities, and (iii) the binding occurrences, which indicate the relations between elementary events that create narratives. A limit of this approach is that the author claims that this grammar cannot be expressed using the standards of the Semantic Web. A narrative ontological model has been developed by the Labyrinth Project and is presented in [29]. The Labyrinth system allows users to explore digital cultural heritage archives and is based on narrative relations between knowledge resources. The Labyrinth project relies on the notion of "cultural archetype", i.e. a core representation encompassing archetypal stories and characters, proposed as a conceptual framework for accessing to archives of heterogeneous media objects. In [52], a review of several attempts is reported, to provide an elementary set of patterns for basic story plots and to describe possible plots in terms of schemas. These schemas may be used to drive another framework for computational story generation (Propper system). An empirical experiment to identify similarities between narratives is reported in [70]. Empirical studies use human participants, where these are presented with triples of stories and are asked to rate the degree of similarity between the stories and the appropriateness of a story as a summary of another story. The results support the position that the two ratings are highly correlated. [106] describes a preliminary approach to create a flexible framework (SONNET) to integrate, select, and reuse narrative models. This framework includes a lightweight ontology language for the definition of key terms and interrelationships among them. It allows specifying model metadata by means of which developers can rapidly discover and understand models. Finally, [77] claim that the concept of narrative does not correspond only to the list of its events but it also includes its *evaluation*, i.e. the interpretation of the sequence of events. Although an intended interpretation exists in the narrator's mind, this interpretation can be accepted, disputed or rejected by the readers (interlocutors). Interpretations indeed change according to the norms and the beliefs of the interlocutors' social belongings.

2.3 Ontology Background

In this Section a brief description on the ontology background is given, in order to provide to the reader a complete overview on the context on which this thesis was developed.

The Semantic Web is defined by Tim-Berners Lee [14] as “...a common framework that allows data to be shared and reused across application, enterprise, and community boundaries”. The data published in the Semantic Web domain are enriched with metadata that specify their semantic context. The data can be processed by computers and “Linked Data” method is used for publishing them to the World Wide Web, where the data can be linked and integrated. Tim Berners-Lee gave a presentation on Linked Data at the TED 2009 conference [5], where he explained Linked Data using three simple principles:

1. All kinds of conceptual things have names starting with HTTP.
2. By exploring these HTTP names, it is possible to retrieve data in a standard format that describe these concepts.
3. Once these data are retrieved, relationships between one concept and another are built.

Linked data use standard web technologies as Uniform Resource Identifiers (URI) to uniquely identify data. Different from an HTML (HyperText Markup Language) Web page, the Web of Data uses a simple meta-model called RDF (Resource Description Framework). The RDF Schema (RDFS) defines triples with special meaning that provide the basic building blocks for implementing hierarchical ontologies. Ontologies, in the computer science sense, are used to represent knowledge and employ hierarchical structures of classes and properties reflecting different levels of specificity (or levels of knowledge) from which inferences can be made. In the follow-up of this Section the definition of ontology is reported along with a brief description of the ontology languages.

2.3.1 Ontology Definition

The word “ontology” has different meanings in different communities. As explained in [57], “Ontology” with upper-case initial refers to a philosophical discipline, i.e. to the branch of philosophy which studies the existence of nature. The first definition of Ontology is reported in Aristotle’s Metaphysics[6], where the author defined it as the science of “wisdom” i.e., the study of attributes that belong to things because of their very nature. Instead, in the experimental sciences, which aim at discovering and modelling reality under a certain perspective, an ontology is defined as the science that focuses on the nature and structure of things independently of any further consideration, and of their actual existence. it is possible to develop an ontology of fictional entities

[5]http://www.ted.com/talks/tim_berners_lee_on_the_next_web

[6]http://plato.stanford.edu/entries/aristotle-metaphysics/

like Cyclops: although they do not have real existence, their nature and structure can be described in terms of general categories and relations.

In Computer Science an ontology is considered a type of information object or computational artefact [55, 58]. In particular, in Artificial Intelligence (AI) the concept of existence is defined as what can be represented. In this context, computational ontologies provide formal models to represent the structure of an AI system, identifying the relevant entities and relations that emerge from its observation. An ontology engineer analyses these entities and organizes them into concepts and relations.

In AI, the core of an ontology consists of a hierarchy of concepts. For example, if someone is interested in representing mankind, Person, Male and Female are relevant concepts and relationships between persons can be relevant relations. In the nineteens, several definitions of ontology were provided by different scholars. Gruber [58] defines an ontology as an "explicit specification of a conceptualization", Borst [16] as a "formal specification of a shared conceptualization". Guarino [56] defines ontology as "a logical theory accounting for the intended meaning of a formal vocabulary, that is, its ontological commitment to a particular conceptualization of the world. The intended models of a logical language using such a vocabulary are constrained by its ontological commitment. An ontology indirectly reflects this commitment (and the underlying conceptualization) by approximating these intended models". In this perspective the conceptualization is intended as a shared view expressed in a formal machine readable format. These definitions were merged in Studer et al., where an ontology is defined as "a formal, explicit specification of a shared conceptualization" [126]. From these definitions it is possible to realise that a conceptualization is in the human mind, and its precise formal structure is not directly accessible.

As described in [21], many experiments to integrate heterogeneous databases were done in 1990s [25, 66, 135]. These experiments have demonstrated that in order to merge two databases, an explicit and formal representation of the meanings of their structures is necessary. As a result of this analysis, the term ontology now indicates the product of an engineering process, that is, a set of concepts and their relationships that allow to formally represent a domain of interest.

The challenge of designing ontology languages can be described as a compromise between the expressive power needed to approximate human conceptualization, and the logics applied in Computer Science. In the simplest form, an ontology consists of a finite list of terms and relationships between these terms. Indeed, common components of ontologies include:

- Individuals: instances or basic entities
- Classes: sets, collections, concepts, classes in programming, kinds of entities
- Attributes: properties, features, characteristics, of the entities (and classes)
- Properties: semantic relations that relate classes and individuals one another
- Function terms: complex structures formed from certain relations that can be used in place of an individual term in a statement

- Rules: statements in the form of an if-then sentence that describe the logical inferences that can be drawn from an certain assertion
- Axioms: assertions (including rules) in a logical form that comprise the overall theory used in the ontology to describe its domain of interest.

The relation typically include hierarchies of classes. In addition to subclass relations, ontologies may include information such as:

- Properties (X works at Y)
- Value restrictions (only research members can teach courses)
- Disjointness statements (research and technical staff are disjoint)
- Specification of logical relationships between entities (for each laboratory exists a director)

2.3.2 Ontology Languages

Ontologies are commonly encoded using ontology languages. In computer science and AI, ontology languages are formal languages that allow encoding knowledge about specific domains and often include reasoning rules that support the processing of that knowledge. Ontology languages are usually declarative languages, and are almost always generalizations of frame languages. They are commonly based on either first-order logic or on description logic. In 2004 the World Wide Web Consortium (W3C) announced its support for two Semantic Web technology standards, RDF (Resource Description Framework) and OWL (Ontology Web Language). RDF [11, 18, 88] is a language used to provide a standard for metadata describing the semantics of information about web resources in a way that is accessible to machines. It is based on a RDF Schema, a "grammar" that defines triples with special meaning that provide the basic building blocks for implementing hierarchical ontologies. The classes and properties defined in RDF Schema (RDF/S) are used in an RDF graph, i.e., a collection of RDF statements representing a "slice of reality". An RDF statement is a triple in the form of subject-predicate-object. An example of such a triple statement would be:
Subject: "http://dbpedia.org/resource/Virgilio"
Predicate: "dc:creator"
Object: "http://dbpedia.org/resource/Eneide"
RDF uses URIs to identify resources and it has an XML-based syntax. In an RDF graph, all triplets are represented by nodes and arcs and are labelled with qualified URIs.

OWL [6, 32, 91, 108] is an extension of RDF/S. In particular, OWL builds on RDF and RDF/S, using XML syntax. OWL uses the same RDF meaning of classes and propertied and has similarities with this language. One difference is that the former has a stronger syntax with more machine interoperability and vocabulary language than the latter. RDF is commonly limited to binary ground predicates, and RDF/S has the limitation that it represents a subclass hierarchy and a property hierarchy. In other words, OWL is more expressive than RDF/S. W3C classifies OWL into three sublanguages: OWL Lite, OWL DL and OWL Full [10].

- OWL Lite is the simplest version of OWL and provides a classification hierarchy and simple constraints; it permits only the expression of relationships with maximum cardinality equal to 0 or 1, thus being designed for easy implementation. In this sublanguage, there is some restriction of OWL DL to a subset of language constructors, with some limitations such as an absence of explicit negation or union. The disadvantage of this sublanguage is restricted expressiveness.
- OWL DL is so called because it uses Description Logic to represent the relations between objects and their properties. Indeed, it provides maximum expressiveness while preserving the completeness of computational properties. OWL DL is computationally decidable, that is, a reasoner always terminates and returns the correct result. Decidability was a core design criterion in the development of OWL DL, and many of the syntactic restrictions of OWL DL have been introduced to guarantee decidability.
- The sublanguage OWL Full provides highest expressiveness and the syntactic freedom of RDF but without preserving guarantees on computational complexity. For OWL Full, computational aspects have never played a role in its design, only logical/semantic aspects. Since none of the restrictions that OWL DL uses for retaining undecidability are introduce in OWL Full, OWL Full can be undecidable.

CHAPTER 3

Research Questions and Methodology

The advent of Digital Libraries (DLs) and the Web, coupled with the increase of the number of tools to automatically create Web sites, have driven cultural institutions (e.g., museums, archives) to undergo a massive digitization process and to publish the resulting digital catalogs online. The main problem of the resulting cultural Web sites, and of their back-end DLs, is the limitation of the informative services offered to the user. Although those DLs rely on broad knowledge bases, they offer simple search functionalities based on free-text queries that return a ranked list of the information objects contained in the DL catalog. No semantic relation among the returned objects is usually reported that can help the user in interpreting the obtained result and, more importantly, in obtaining a more complete knowledge on the subject of the research. The main reason for this behavior usually lies in the low level of formality at which the knowledge contained in the DL catalogs is expressed. Such knowledge is mostly embodied in texts, images, or in multimedia objects, and extracting knowledge from these media is a notoriously difficult task. The situation is very similar to that of the web, but with an important exception: web pages are self-contained descriptions, and as such each one of them may potentially satisfy the user information need. Moreover, they are hyper-linked to pages describing related entities, and these links are exploited by web search engines to bring the most relevant pages up in query results.

The introduction of the Semantic Web [12], and in particular of Linked Data [14], has the potential of improving the search functionalities of DLs. In fact, thanks to the languages and tools developed in the context of the Semantic Web, many cultural institutions have moved to represent their vocabularies as formal ontologies and their metadata as formal descriptions couched in terms of those ontologies. This move has been achieved by means of formal languages such as RDF, used for metadata records, and such as RDF Schema or OWL, used for formal ontologies. The resulting new generation DLs allows implementing more effective search functionalities, due to the richer formal representation of the semantics of the objects described in their catalogs.

The new search functionality that this thesis envisages aims at exploiting this richness to overcome the above mentioned problem. According the our vision, a search query should not return just a ranked list of objects; rather, whenever possible, it should present a narrative to the user, based on the objects of the library that are relevant to the query and on a set of semantic relations that connect these objects into a story that is meaningful to the user. In addition, the use of narratives in DLs can be useful for two aspects: (i) for modelling and managing the provenance of the represented knowledge and of the objects in a DL; and (ii) for enriching of the content of the DL with events that connect the DL objects into meaningful semantic units.

For what concerns the provenance, narratives could be important (i) to improve the access to the DL objects; and (ii) for the preservation of such objects. The present provenance models are models with a very limited amount of formally represented semantics. In order to be general, such models end up using concepts such as process and states [101] that may be very helpful at the implementation level, because they can be applied to capture the provenance of any object; but offer very little help to the curators, who need to talk about specific types of process and state. As a result, the burden of mapping the actual processes taking place in each domain into the notion offered by these models, rests entirely on the curators themselves, partially defeating automation. The idea of this thesis is that the DL curators should be given more effective tools in order to represent the provenance of the objects they curate, and that narratives can be such a tool. In fact, the representation of provenance as a story is very intuitive because it directly reflects reality. Moreover, the types of events and actions that can occur in the provenance of the objects in a certain domain, can be easily identified, along with their parameters and along with the relationships that connect them. This implies that curators can be equipped with a tool for creating narratives endowed with a predefined set of event and action types from which they can select for composing the provenance models of their curated objects. Such a tool would be much easier to use than current models, and as such would greatly improve the effectiveness of digital object curation.

For what regards story generation systems for content enrichment, currently DL search functionalities are principally centred on metadata: a user query returns a ranked list of results, without specifying the relations among them. An example of a list of results is reported in Figure 3.1, which shows the output of the Europeana[1] search functionality[2] for the query *Versailles Conference*.

One of the proposals promoted in this thesis is that the DL should return a semantic representation of the relations among the items of the resulting list, using a narrative approach. At the same time, the digital curator should be able to create the narratives for the DL objects (s)he would like to promote. For example, a narrative about the Versailles Conference can be defined by a digital librarian linking the information objects of the Versailles treaty (e.g. photographs of the people, the final declaration etc.) to an event which represents the Conference. Such event may be further divided into subevents to be placed on a temporal axis. A graphical example of this idea of a narrative about the Versailles Conference is reported in Figure 3.2.

[1] http://pro.europeana.eu/web/guest/home
[2] http://eculture.cs.vu.nl/europeana/session/search

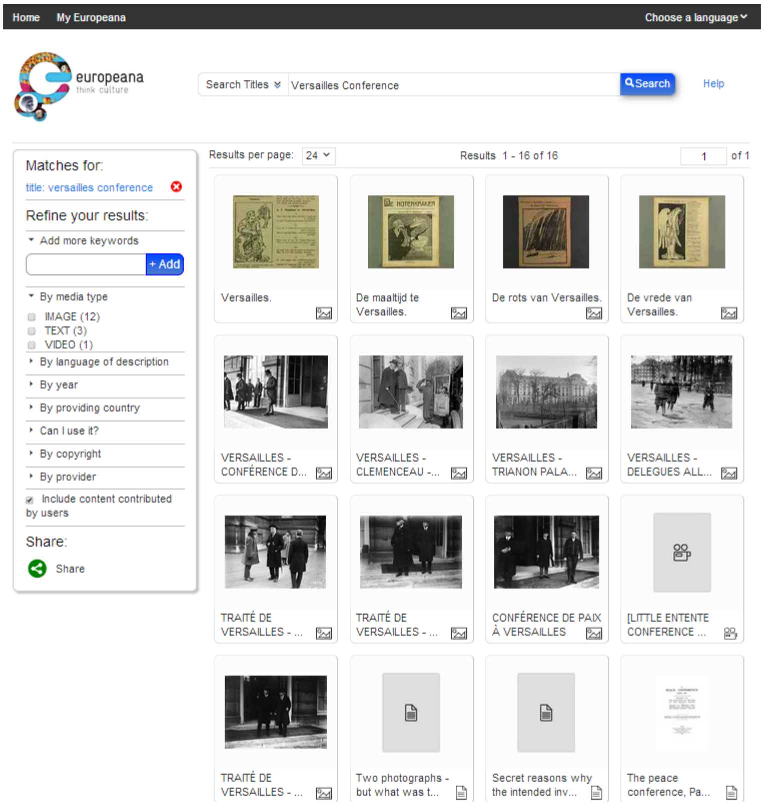


Figure 3.1: *The output of Europeana for the query* Versailles Conference.

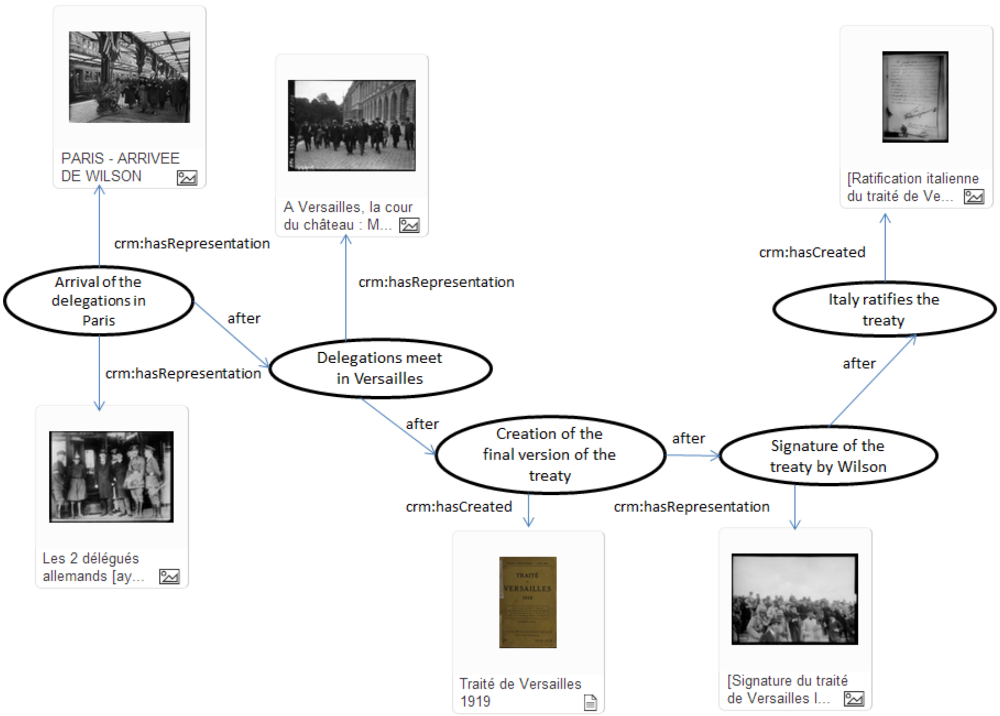


Figure 3.2: *An example of narrative for the* Versailles Conference.

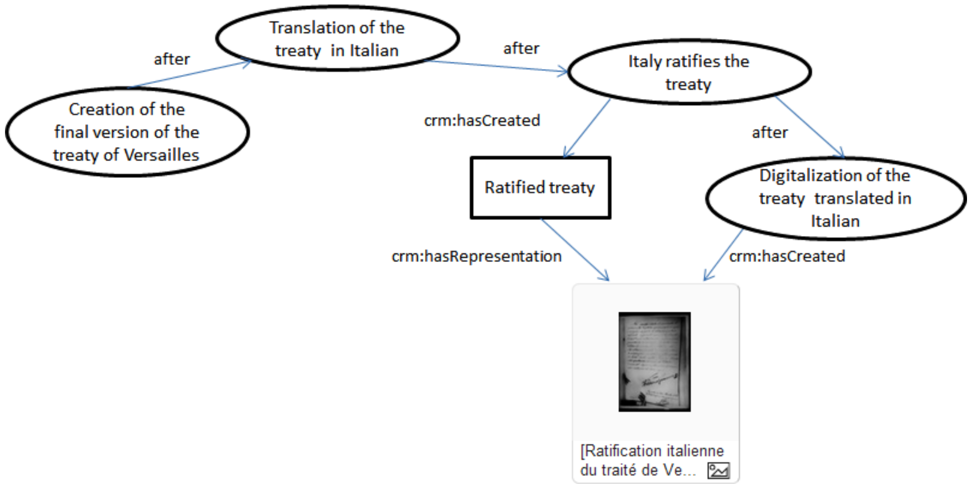


Figure 3.3: *An example of narrative of the provenance for the* treaty of Versailles ratified by Italy.

The main event of the Conference can be divided in several sub-events, to which a digital object belonging to the Europeana DL is associated. To illustrate the idea in an easy way: (i) the temporal relation *after* defined by Allen [4]to link an event to another was used; (ii) the relations crm:hasRepresentation and crm:hasCreated, defined in the CIDOC-CRM ontology [36] to link the events to the digital objects included in Europeana.

Figure 3.3 shows the narrative about the provenance of the *treaty of Versailles ratified by Italy*. The main event is divided into sub-events which reconstruct the provenance of the digital version of the treaty included in Europeana.

The DL could serve both narratives as output of a query about the Versailles Conference, thereby providing the user a semantic network telling a story. The same narratives could be re-used also in the context of a bigger narrative, for example one regarding the World War I.

3.1 Methodological Approach

The methodological approach proposed in this thesis to introduce narratives as a new functionality in DLs is very similar to the one that characterizes a common workflow to develop an algorithm in computer science. Figure 3.4 reports the main phases of this workflow.

The phases algorithm development were elaborated in order to adapt them to our aim. In particular, the adopted methodological approach consists of the following phases:

- The identification of the case studies to define the problem and the possible solutions. In particular, the identified aim of this thesis is the formal representation of a special type of narratives: the biography. This choice is motivated by the fact that this study is partially supported by the Italian national research project “Towards a Digital Dante Encyclopedia” (2013-2016), in the context of which Dante’s experts and biographers collected data about life’s events of the poet.

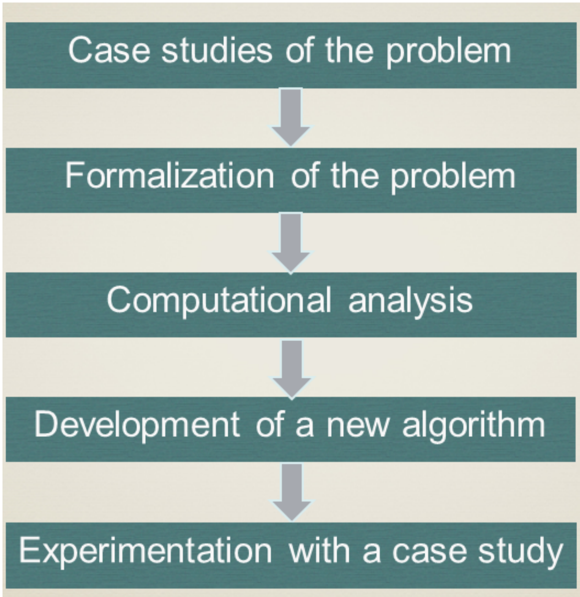


Figure 3.4: *Common workflow to develop an algorithm.*

- The creation of a conceptualization of the problem, in which the issue is described and analysed in its main parts.
- The development of a mathematical specification of the conceptualization, as a pre-requisite to axiomatize the conceptualization in a formal ontology.
- The creation of an ontology for encoding the mathematical specification, thereby representing the meaning of complex narratives, in a formal adequately detailed and computer-understandable format, using the CRM standard ontology as reference vocabulary.
- The experimental implementation of the ontology using a Dante's biography as case study.

The workflow used for this thesis is shown in Figure 3.5.

In order to populate the ontology of narratives, a semi-automatic software tool was developed. The tool allows the user (i) to encode the original information with the OWL language, and (ii) to exploit the collected knowledge by questioning and inferencing, giving also an automatic visualization of the data in the form of events on a timeline, tables and charts. Before explaining the details of the knowledge representation and of the population and visualization systems, the analysis of the requirements is reported to define the practical goals of this research.

3.2 Research Requirements

As explained in Section 2.3 and defined in Computer Science and Information Science [57], an ontology is a formal naming and definition of the types, properties, and interrelationships of the entities that really or fundamentally exist for a particular domain of the discourse. For interoperability purposes, an ontology is usually created and

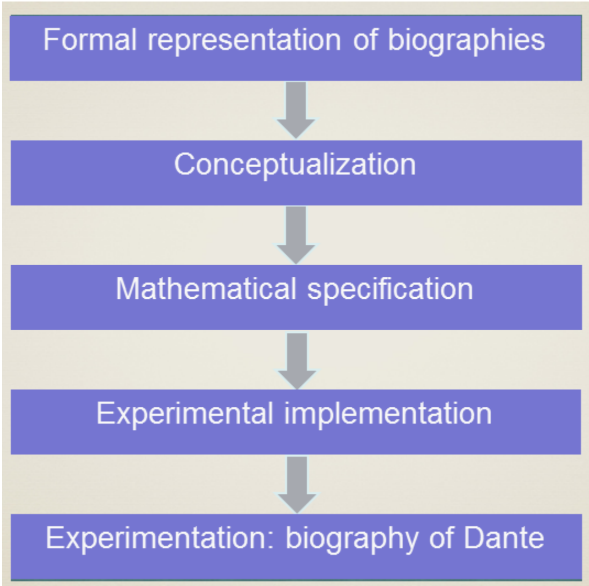


Figure 3.5: *The workflow used in the project implementation.*

exchanged in a Semantic Web language. Then, it is used as the conceptual backbone of an information system, whose information base consists of a representation of a particular slice of reality built with instances of the classes and the properties defined by the ontology. The information system serves the needs of a community of users, therefore analysing users' requirements is an integral part of information systems design and it is important for the success of interactive systems. As specified in the ISO 13407 standard (ISO, 1999)[3], understanding the needs and requirements of the users is the first step to develop successful systems and products. Indeed, the result of this analysis can bring a project an increase of productivity, a better quality of the work, smaller costs for providing support and training, and improvement of users' satisfaction [37].

The first step in users' requirements analysis is to collect background information about the users and the processes that currently take place, through structured interviews. Using this approach, an initial set of requirements was developed, in particular for four phases of the project: (i) creating an ontology for representing the knowledge on narratives; (ii) developing of a tool to populate the ontology; (iii) defining the knowledge to extract from the created knowledge base; (iv) developing a web interface for visualizing narratives in easy and user-friendly way. The preferred users for this research are scholars who want to create and access narratives about the life and the works of the authors they study. Based on the results of the data collected by interviewing the scholars, and on the study of the literature, principal users' requirements were identified. These are reported below for each phase:

1. Phase 1: Creating an ontology for representing the knowledge on narratives, using the Semantic Web technologies.

    - Representing the events that compose the narrative in a chronological order
    - Reconstructing the plot of the narrative, i.e. as the events are narrated

[3] http://www.iso.org/iso/catalogue_detail.htm?csnumber=21197

- Representing the provenance, i.e. the inferential process of the narrator who composes a narrative from primary sources whose contents are propositions about the events of the narration

2. Phase 2: Developing of a web tool to populate the ontology. The tool should allow:
   - Creating the events the compose the narratives in an assisted way
   - Defining the factual components that characterize the events (person, place, time, physical objects and conceptual objects)
   - Identifying the roles for person entities
   - Linking a digital object to an event (e.g. digital objects included in Europeana)
   - Defining the type of each event choosing from a list of pre-defined options
   - Defining the primary sources of each event and a list of the main bibliographic references
   - Linking the events to the appropriate semantic relations
   - Storing the textual fragment representing an event
   - Relating the textual fragment with its narrator
   - Storing the primary sources from which the events are defined
   - Saving the narrative as a digital object

3. Phase 3: defining the knowledge to extract from the created knowledge base:
   - Events along with their primary sources
   - Events with particularly related entities (e.g. place, person)
   - Events happened in a defined range of time
   - Events linked to particular relations (e.g. causal or mereological)

4. Phase 4: Developing a web interface that allows:
   - visualizing a narrative on a timeline
   - visualizing events (all or only someone defined by the user) along with their primary sources in format of table, exportable in CSV format
   - visualizing events happened in a specific range of time (defined by the user) in format of table, exportable in CSV format
   - using network graphs to visualize an event and its related entities
   - using network graphs to visualize a particular entity and its related events
   - using network graphs to visualize the different types of relations that connect events

During each of these four phases, a prototype to illustrate the satisfied users' requirements was developed. Feedback from users of the prototypes was used to validate the solution and refine the requirements.

CHAPTER 4

An Ontology for Representing Narratives

In order to introduce narratives in digital libraries, the present chapter presents a narrative conceptualization, in the sense of the development and the clarification of the concepts that are relevant for narratives. Such conceptualization builds on the vast amount of research conducted thus far on narratives, as accounted for in Chapter 2. More specifically, the conceptualization relies on fundamental notions developed in Narratology, resting upon the basic entities borrowed from Artificial Intelligence, and the Event Calculus in particular.

A formalization of the conceptualization, in the sense of a mathematical specification, is then reported, aimed at giving a precise definitions of the notions in the conceptualization. The formalization is subsequently expressed, or rather approximated, at the linguistic level, so obtaining the sought ontology. The expression relies on:

1. the Web Ontology Language[1] (from now on OWL for brevity), a logical language of the Semantic Web family, providing a machine readable-syntax for the reference notions and for the axioms capturing the formal semantics of these notions. The OWL language was chosen for technical interoperability reasons, looking at Linked Data and at the Web as the ideal medium and the ideal infrastructure for producing and consuming narratives.
2. the CIDOC-Conceptual Reference Model (from now on CRM for brevity) foundational ontology, providing the basic vocabulary to express the classes and properties of the conceptualization. The CRM was chosen for two reasons: from one hand, it is one of the most widely adopted ontologies in the domain of Cultural Heritage, where both digital libraries and narratives belong. From the other hand, CRM is an ISO standard since 2006 (ISO21127:2006) and renewed 2014

[1] `https://www.w3.org/TR/owl-features/`

(ISO21127:2014). Both these factors are crucial to attain semantic interoperability, based on sharing existing ontologies. Semantic interoperability is a two-way concept: on the one hand, it indicates the aim of widening the usage of the ontology for narratives by making it re-usable; on the other hand, it indicates the aim of re-using existing ontologies as much as possible when developing a new ontology.

Then, as it will described in Chapter 5, the so obtained ontology is used to validate the conceptualization on a realistic case study: the biography of Dante Alighieri, the major Italian poet of the late Middle Ages.

It is useful to stress that a full ontology for narratives is at the moment a too ambitious goal, given the fact that the very notion of narrative is still under debate in a number of fields, including literary studies, linguistics, cognitive science and artificial intelligence. The work presented here is therefore to be understood as a first proposal towards the introduction of narratives in Digital Libraries, to be further amended and improved as more insights are gained through experimentation. An elaboration of such further developments is presented in the Conclusions (8).

The present Chapter is structured as follows: In order to derive the conceptualization, the Event Calculus (EC for brevity) is first analysed in Section 4.1. The EC is one of the most studied and widely known formal theories in Artificial Intelligence aiming at modelling the dynamics of intelligent systems. As such, the EC embodies a representation of some of the basic components of narratives, such as time, events and situations. Section 4.2 presents the conceptualization introducing a formal computable representation of narratives in an informal way, based on the narratology background reported in Chapter 2 and on the basic notions reviewed in the Section 4.1. Finally, in Section 4.3 a mathematical specification of the conceptualization is given.

4.1 Basic Notions in Formal Narratives

This Section recapitulates the definitions of the basic notions of narratives, as found in Artificial Intelligence, starting with the Situation Calculus and then moving to the Event Calculus. These two *calculi* are by far the most studied ones, and their development has seen the flourishing of dozens of variations or extensions, each capturing a particular intuition or coping with some problematic aspect. They aim at representing and reasoning about dynamic systems typically dealt with by robots, and although the flux of events, actions and changes in narratives typically follow quite different rules from those at work in such dynamic systems, these calculi deal with the very same basic notions, and offer a strong formalization of them. So they are a natural place where to start the investigation.

The Situation Calculus (SC) is a logical language for representing and reasoning about dynamical domains [89, 90]. In dynamical domains the scenarios change because of the actions performed by the agents. A dynamic world is modelled as a series of situations resulting from actions performed in the world. SC represents changing scenarios as a set of first-order formulae, sometimes enriched with some second-order features [131]. The basic elements of the calculus are:

- Situations. A situation is a state of the system under study resulting from a sequence of actions. Actions are represented using the function symbol *do*, so that *do(a, s)* is the situation resulting from the execution of action *a* in situation *s*.
- Fluents. Fluents are functions and predicates that vary over situations (*e.g.,* the location of an agent typically changes depending on a specific situation). Fluents are used to describe the effects of actions. Fluents can be distinguished in relational fluents and functional fluents. The formers can only be true or false in a given interpretation, whereas the latters can take a range of values.
- Actions. Actions are changes performed by agents when moving from a situation to another in a dynamic world. Each action can be described in the simplest version of Situation Calculus using two axioms: (i) the Possibility Axiom that specifies when the action can be performed; and (ii) the Effect Axiom that defines the consequences of executing the action.

SC works well when there is a single agent performing instantaneous, discrete actions. When actions have duration and can overlap with each other, an alternative formalism is the Event Calculus (EC) [69, 100, 102]. The EC is a formal theory for reasoning on actions and changes, and is based on time points rather than on situations. EC allows reasoning over intervals of time and over fluents that are time-dependent rather than situation-dependent. EC axioms define a fluent true at a point in time if "the fluent was initiated by an event at some time in the past and was not terminated by an intervening event" [118]. Davidson [31] defines *actions* as a sub class of *events*. In Davidson's opinion, the distinct sign between general events and actions is the intentionality of actions, which n turn implies the presence of an agent performing the action. The EC has actions, in the same way as SC. However, Davidson's distinction between events and actions is not present; in the EC, actions are events. In the following list, the logical definitions of the components of narratives that are interesting to represent narratives is reported.

- *Generalized events*. In the context in which actions and objects are aspects of a physical universe with a spatial and temporal dimension, a generalized event is a space-time chunk. This abstraction allows thinking to generalized events concepts like actions, locations, times, fluents and physical objects.
- *Processes*. Processes are events included in the subcategory of generalized events that have a clear beginning, middle, and end. Processes can also describe states referring to processes of continuous non-change. It is possible to combine states and events using the fluent calculus approach. The fluent calculus combines primitive states and events. For example the event of two things happening at once is denoted by the "Both" function: Both(e1, e2) is often shown in abbreviated notation as e1 and e2.
- *Time*. Another subcategory of Generalized Events is constituted by time intervals. It is possible to identify two types of intervals: moments and extended intervals. The distinction between these two types is that moments have zero duration. To represent moments and extended intervals, a time scale is needed along with points on this scale. Start, End, Time and Duration functions are used for the representation, where:

- Start, End: earliest, latest moments of an interval respectively
- Time point: an element of the time scale
- Duration: difference between end and start times

In order to reason on time intervals, it is possible to use the predicates defined in the Allen's calculus [4], e.g. Before, Equal, Meets, Overlaps, During, Starts, Finishes. Alternatively, it is possible to adopt a different reasoning that just relies on time points, totally ordered by the time precedence operator.

- *Physical Objects*. A physical object is a space-time chunk and thus it can be considered as a subcategory of Generalized Events. It is possible to describe the changing properties of objects using fluents.
- *Mental events and mental objects*. A mental event is a particular occurrence of something going on in the mind of an agent. In particular, agents can have beliefs and deduce new beliefs. The relations between an agent and mental objects such as propositions, like believes, knows and wants are named "‘propositional attitudes"’ because they identify attitudes that agents can have towards a proposition [118]. Propositional attitudes are the kinds of mental events that are relevant in the present context. Using the reification method it is possible to turn a proposition into an object that could become an argument of a sentence, because only terms (not sentences) can be arguments of predicates.

4.2 Conceptualization

This Section introduces a formal computable representation of narratives in an informal way, based on the narratology background reported in 2.1 and on the basic notions reviewed in the previous Section. In particular, the idea is that a *narrative* consists of three main elements:

1. the *fabula*, directly representing the fabula as defined by the Russian formalism, i.e. the sequence of the events that composes the story, each with its own features, in chronological order;
2. one or more texts that narrate the fabula, called *narrations* and that correspond to the Bal's definition of *presentation*;
3. a *reference* function that connects the narrations to the fabula and allows to derive the *syuzhet* (or plot) as defined by the Russian formalism.

Note that all these three elements are relevant in a Digital Library (DL), because a DL typically contains both forms narratives, narrations (i.e., digital representations of texts) and fabulae, the latter giving an explicit representations of the semantics of the former. Moreover, texts in a DL are often structured in a way that isolates textual components and textual fragments therein, allowing the expressions of the reference function that connects narrations to fabulae. Hence, fabulae, narrations and plots are central notions in a DL, and the ontology accounts for that centrality by considering these notions as first-class citizens. In the following each of these three elements is analysed in detail.

4.2.1 Fabula

The fabula consists of a set of events, an event being an action or occurrence taking place at a certain time at a specific location. As already pointed out, in EC the terms Actions and Events are interchangeable and represent changes performed over time, whereas Davidson defines actions as a particular kind of events, namely events performed by agents driven by intentions. This thesis subscribes to Davidson's view, and considers actions as a sub-class of events. The events in a fabula may be (a) real, such as those directly witnessed by a scholar, and recorded in a DL for communication purposes, (b) hypothetical, such as those recorded by a historian in the process of reconstructing a particular piece of history, or (c) fictional, such as those created by writers in the literature. The ontology does not commit to any particular kind of event, but it commits to law of nature of our reality. Therefore it applies only to fictional events that commit to the same laws, i.e., reality-like ones. If real and fictional events are mixed in the same fabula and the application requires distinguishing the former kind from the latter, it is the responsibility of the user of the ontology to introduce the necessary mechanisms to realize the distinction; one way of doing that would be by introducing two sub-classes of the event class, one for fictional events and the other for real events. The events in a fabula are connected to each other by three kinds of relations: mereological, temporal and causal relations. Each of these is dicussed in a separate Section below.

Mereology A primitive mereological relation was assumed, called *direct part-hood*, relating events to other events that include them as *direct* parts, that is without any other intermediate event. For example, the birth of Dante Alighieri can be a direct part of the youth of Dante, and the youth of Dante can be a direct part of the life of Dante. Clearly, the selection of events and of their direct parts is a choice of the application; what counts as a direct part in an application, may be not such in a different application. If an event e is a direct part of an event e', it can be said that e is a direct sub-part of e' or that e' is a direct super-part of e. Like any mereological relation, direct part-hood is acyclic, since no two events can be a direct part of each other, either directly or through any number of other events. In addition, it is assumed that an event can be direct part of at most one other event, thereby excluding fabulae in which the same event is a direct part of two different events. This last restriction is probably less cogent than acyclicity, however it introduces a computationally manageable atomization of events that can always be achieved, even in cases where an event e is in reality seen as a direct part of two other events e_1 and e_2. In this case, a third event e' is introduced, which consists of the common happenings of e_1 and e_2 and includes e as a direct part, while e_1 and e_2 are reduced to two events e'_1 and e'_2, obtained by excluding e' from e_1 and e_2, respectively. Clearly, e'_1, e'_2 and e' are not related by any direct part-hood relation; as such they are said to be pairwise *incomparable* with respect to direct part-hood. Technically, the above considerations make direct part-hood a partial, acyclic function over the events of a fabula. Graphically, direct part-hood can be visualized as a set of trees, each of which was called a *part-hood tree*. The roots of the part-hood trees are *maximal* events with respect to direct part-hood, that is pairwise incomparable events that are not direct parts of other events in the same fabula. On the other hand, it is often useful to be able to speak in general terms about the "events that are included" in another event e, meaning all the events that are either direct parts of e, or are direct parts of the direct parts of e, and so on, up to a finite but arbitrary level of part-hood nesting. For example, one may speak about the events in Dante's life, such as Dante's youth and Dante's birth. In order to support these expressions, a second part-hood relation was introduced, called *non-direct part-hood*, or simply *part-hood*. Intuitively, part-hood does not add any new mereological knowledge on the events of a fabula, since it is just built on direct part-hood. In fact, it turns out that from a technical point of view part-hood is the reflexive, transitive closure of direct part-hood. As such it is a reflexive, antisymmetric and transitive relation, i.e. a partial order over the events of a fabula. Each part-hood tree is transformed by the transitive closure into an upper semi-lattice, having the tree's root as a maximum.

Time Temporal relations place the events in a fabula in a specific temporal position with each others. It was assumed time to be an infinite countable set of time points, totally and strictly ordered by the relation of temporal precedence. As a strict order, temporal precedence is an irreflexive and transitive relation, hence asymmetric. The temporal equality relation over the time points, in order to be able to model events that co-occur in time, was also introduced. The combination of the two relations gives a total order over the time points. In order to relate events and their temporal occurrences, in the conceptualization two total functions over the events of a fabula are introduced:

- the *event beginning* function, associating an event with the time point at which the event begins; and
- the *event ending* function, associating an event with the time point at which the event ends.

Obviously, these functions must satisfy the condition that, for any given event, the beginning of the event always precedes or co-occurs with the ending of the event. In ordinary speech, these relations are often directly stated between the involved events, e.g. it is stated that Dante's life starts with Dante's birth. These statements can be understood as abbreviations of the more involved statements making proper usage of the relations; in the example, the correct statement is that the beginning of Dante's life co-occurs with the beginning of Dante's birth.

Causality A single primitive causal relation was assumed, relating events that in ordinary discourse are predicated to have a *cause-effect* relation in the narrator's opinion, e.g., the eruption of the Vesuvius causing the destruction of Pompeii. It is important to notice that the accurate modelling of the causal relationships that connect the events in a physical or chemical process is not an objective of a very specific kind of narratives, those found in science. The scope of the present thesis is much wider, including fabulae whose events may be years apart in time (or centuries, like in history) and the causal connection between such events may be indirect, that is established through other events which may be unknown or not relevant, hence not directly included in the fabula. For this reason, the primitive causal relation of the conceptualization can be thought of as a generalization of scientific causality, produced by the reflexive, transitive closure of the atomic relationships that constitute scientific causality. Therefore, such primitive relation was called as causal *dependency* and model it as a reflexive, transitive relation over the events of a fabula.

Interactions The three main relations defined on events interact with each other. There are three possible kinds of interactions, depending on the involved relations:

1. direct part-hood and temporal relations,
2. direct part-hood and causality, and
3. temporal relations and causality.

The interaction between direct part-hood and temporal relations is straightforward: the period of occurrence of an event is included in that of the direct super-part of the event, if any. The interaction between direct part-hood and causality is less obvious. Let e be a cause (in the sense defined above) of e' and let consider the direct sub- or super-parts of e. Any direct part of e does not necessarily cause e'. For instance, if the move of the wooden horse inside Troy caused the destruction of the city, is it not correct to conclude that any sub-event of the move, such as the opening of the gates, caused the destruction of the city. On the contrary, it is correct to conclude that any direct super-part of e has caused e', because any such super-part would include the "real" cause, that is e. To continue with this example, assuming the war of Troy is a direct super-part of the move of the wooden horse inside Troy; since the latter caused the destruction of Troy, it seems correct to conclude that also the former did, in a more general sense. Consider now direct sub- or super-parts of the caused event, that is e'. It would seem correct to conclude that any direct part of e' is also caused by e; if the move of the wooden horse inside Troy cause its destruction, then it also caused any part of the destruction, such as the destruction of the individual buildings in the city. On the other hand, it is certainly not correct to conclude that what causes an event also causes the direct super-part of it. The fact that the move of the wooden horse caused the destruction of Troy does not imply that it also caused the whole Trojan War, which is a direct super-part of the destruction of Troy. Finally, the only interaction between temporal relations and causality is a negative one: if an event begins after the end of another event, the former cannot have possibly caused the latter; conversely, if an event causes another event, then the former it cannot have possibly started after the latter's end.

4.2.2 Narration

Each narration of a fabula consists of a narrator and a text, the former being the *author* of the latter. The notion of narrator that was assumed is very wide, including individual persons, such as Dante, more than one persons, such as the novelists Maj Sjöwall and Per Wahlöö, or complex bodies, such as the Italian State. The modelling of literary text is an active field of investigation at the crossroads of many disciplines, aimed at highlighting certain features of a text. For the present purposes, the only aspect that is functional to the modelling of narratives was focussed, namely *textual content*, which is the language expression that constitutes a piece of text. Technically, such language expression is a sequence of finite length consisting of characters drawn from the narration alphabet, that will be called "text chunk". Therefore the text of a narration will be modelled as a text chunk, thereby adopting a purely extensional, purposefully minimal view of text.

A text is typically divided by the narrator in meaningful units, that collectively form the *structure* of the text. It is commonly agreed that structure is a substantial part of a text, so it must be represented in a DL in order to offer a proper fruition of the narration to the users of the DL. Since an extensional view of text was adopted, the structure of a text can be simply modelled as the set of the continuous textual units created by the narrator upon writing the text. Each such unit is a text chunk, so the direct parthood relation between the text chunks in a narration-establishing, e.g. the chapters in a book or the sections in a paper, is simply the string containment relation between the text chunks in the structure of the narration. Not any set of text chunks is a structure, though. To qualify as a textual structure, the chunks in the set must satisfy the following conditions:

1. there is a unique chunk, the *root*, which is a strict superset of each one of all the other chunks;
2. the maximal sub-sequences of each chunk are a partition of the chunk.

The first condition guarantees that the whole narration text is part of the structure, while the second guarantees that each chunk of the structure is properly divided into disjoint sub-elements that entirely cover it. Any chunk that has no sub-elements is called a *leaf*. Note that the leaves are a partition of the root. Any chunk that is not the root nor a leaf is called a *non-terminal*. In addition to the structure of a narration, it was needed to model the portions of the narration text that narrate events. Such portions, which was called *narrative fragments* or simply *fragments,* are the domain of the reference function (see next Section below) and as such they play an important role in the conceptualization. Intuitively, a fragment consists of a non-empty set of disjoint text chunks, since the narration of an event may be suspended and resumed an arbitrary number of times by the narrator in the narration text. The term *chunk-set* will be used to designate a finite, non-empty set of disjoint text chunks from the same narration, and the basic properties that a chunk-set S must possess to qualify as a fragment were investigated. Two intuitive conditions ware factored out: the first, obvious one is that S narrate an event and nothing else; the second is that there exists no other chunk-set in the same narration that is strictly larger than S and that narrates the same event and nothing else. Technically, the fragments of a narration were defined to be a set of chunk-sets, each of which narrates an event and is *maximal* with respect to containment (more precision would be necessary to define how a chunk-set contains another chunk-set; however, this specification is a tedious exercise that would make the model unnecessarily complex without adding any insight; so any further condition on fragments was imposed). The everyday operation of a DL requires a substantial amount of information (or *metadata*) about the texts that the DL manages. However, the present purpose are to identify the concepts of narration that are relevant, so all aspects of narrations other than those relevant to the purposes of this research will be ignored.

4.2.3 Reference

Generally speaking, the reference function connects the fragments in a narration to narrated events in the fabula. This connection may be established in different ways, and below the possible alternatives are discussed and the one that best suits the purposes of this research is selected. A preliminary question that must be addressed is how many events a fragment can narrate; technically this amounts to establish whether the reference function is a function in the mathematical sense, or it is a mathematical relation. Based on the experience collected in this study and in similar researches, it is possible to define fragments in such a way that each fragment talks about a single event. However, if that event is part of another event, then it is correct to say that any fragment narrating the former *also* narrates the latter. To exemplify, a fragment narrating the birth of Dante also narrates the life of Dante. In order to reflect the acquired experience as well as this last observation, the same approach as for event mereology was followed, and in the conceptualization a *direct reference function* that associates each fragment,in a narration to an event in the fabula was introduced. Based on this direct reference function, next the *reference function* as a relation including a pair (f, e) if and only if fragment f directly refers either event e or an event that is part of e was introduced. Correspondingly, the term *direct reference* is used when discussing the direct reference function, for instance when saying that a fragment directly refers to an event, and the generic term *reference* is used when discussing the just introduced reference function, for instance when saying that a fragment always refers to a maximal event. Once established that direct reference is a function, its properties were discussed. Specifically:

- *totality*, i.e., whether direct reference is defined on every fragment. It is need to recall that a fragment is required to narrate an event; however, this does not imply that a fragment *directly* narrates an event. Intuitively, the reason to define a fragment is that it stands out with respect to the other fragments because of its narrative content, therefore that direct reference was imposed to be total.

- *injectivity,* i.e. whether two fragments can directly narrate the same event. This is indeed not possible, due to the definition of fragment, which requires a fragment to be a maximal chunk-set that narrates an event. As a consequence, if a fragment narrates two events, there is necessarily some text in the fragment that narrates only one of the two, because otherwise the two events would not be distinguishable from one another. But then, the fragment would violate the first condition, asking that a fragment narrates an event *and nothing else*. Therefore direct reference is injective. It is worth to note that the properties of the mereology of events imply that there is a single maximal event indirectly narrated by a fragment, and that all the events narrated by a fragment form a chain from the directly narrated event to the unique maximal event indirectly narrated

- *surjectivity,* i.e. whether *every* event in the fabula is narrated by a fragment. Intuitively, there would be no reason to include an event in the fabula that is not narrated by any fragment in a narration of the fabula. However, a narrative may include more than one narration, and it is intuitively reasonable that some of the narrations in the same narrative miss some events in the fabula (e.g., not all historians agree that Dante went to Bologna in 1308, so some narratives of Dante's life will include that event, while others will not). For this reason, the surjectivity of direct reference is relaxed for a weaker condition, requiring that for each event of the fabula, there exists a narration that narrates that event, that is, a narration that directly refers that event.

Notice that the reference function allows deriving the plot of the narrative. Indeed, by visiting the text of the narration in its natural reading order, it is possible to access the fragments and, via these, the events in the fabula, *in the order established by the narrator*, which may be different from the chronological ordering in the fabula. Finally the interactions of the reference function with temporal relations and with causality was considered. The question is whether the fact that a fragment directly refers a certain event has implications on the events placed in a certain temporal or causal relation with that event. The answer is clearly "no" in both cases, so it is possible to conclude that the reference function is independent from temporal relations and from causality.

4.3 A Mathematical Specification of the Conceptualization

In this Section a specification of the above conceptualization in mathematical terms is provided. The choice of mathematics allows concentrating on the proper capturing of the notions highlighted above, postponing any language consideration at a later stage, once the mathematical specification will have brought forward the required machinery. As it will be shown, some elementary notions of set theory (see for instance [30]) will suffice for the purposes of this research. The starting points are three disjoint countable sets:

- events, denoted as E, members e, e_1, e_2, ...
- time points, denoted as T, members t, t_1, t_2, ..., totally ordered by the time precedence relation $\leq$.
- texts, given by the strings of finite length over an alphabet S, $S^\star$, members s, s_1, s_2, ...

A fabula f is a 5-tuple $\mathsf{f} = \langle E_\mathsf{f}, p_\mathsf{f}, b_\mathsf{f}, d_\mathsf{f}, c_\mathsf{f} \rangle$ consisting of:

- A finite set of events, $E_\mathsf{f} \subset \mathsf{E}$
- The *event composition* function $p_\mathsf{f} : E_\mathsf{f} \to E_\mathsf{f}$ associating *some* event e_1 in E_f with a different event e_2 in E_f, such that e_1 is a part of e_2. In this case, e_1 is a sub-event of e_2 or that e_2 is a super-event of e_1.
- The *event beginning* function $b_\mathsf{f} : E_\mathsf{f} \to T$, associating each event e in E_f with a time-point $t = b_\mathsf{f}(e)$ in T, such that event e starts at time $b_\mathsf{f}(e)$.
- The *event ending* function $d_\mathsf{f} : E_\mathsf{f} \to T$, associating each event e in E_f with a time-point $t = d_\mathsf{f}(e)$ in T, such that event e ends at time $d_\mathsf{f}(e)$.
- The *causal dependence* relation $c_\mathsf{f} \subseteq E_\mathsf{f} \times E_\mathsf{f}$, such that $e_1, e_2 \in c_\mathsf{f}$ if and only if event e_2 causally depends on event e_1.

For simplicity, subscripts from fabula components will be omitted, when there is no ambiguity.

For each event $e \in E$, the pair $(b(e), d(e))$ is said to be the *period of occurrence of* e.

A well-formed fabula is a fabula satisfying the following conditions:

1. The event composition function p is acyclic, so that no event can be, at the same time, a sub-event and a super-event of some other event. Technically, acyclicity can be expressed as the condition that the transitive closure of p, $p^\star$, be an irreflexive relation.

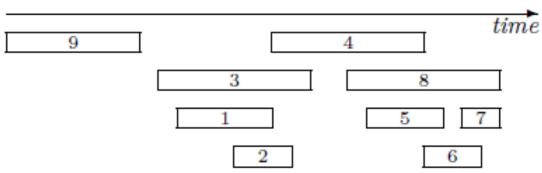
2. No event finishes earlier than its beginning: for each event e in E, $b(e) \leq d(e)$.

3. The period of occurrence of a sub-event is always included in the period of occurrence of its super-event: for each event e in the domain of p, $b(p(e)) \leq b(e)$ and $d(e) \leq d(p(e))$.

4. Causal dependency is a reflexive and transitive relation.

From now on, only well-formed fabulae will tacitly considered.

Notice that events in the same fabula to overlap in time in an arbitrary way are allowed, enabling even the sub-events of the same event to do so. Also, any other condition on causal dependency other than the obvious reflexivity and transitivity was not placed.



level i	Units in $\sigma(i)$
1	[1,2000]
2	[1,1500] [1501,2000]
3	[1201,1500] [1901,2000]

Figure 4.1: *Events in a fabula (left) and the structure of a narration (right)*

Figure 4.1 left gives a pictorial representation of a fabula consisting of nine events, identified with the first nine positive integers, each represented by a rectangle whose horizontal extension gives the temporal extension of the event on the time scale depicted at the top of the figure. The event composition function is depicted by placing sub-events immediately below their super-events. As it can be seen, events 4 and 9 do not have any sub- or super-events; event 3 has 1 and 2 as sub-events, overlapping with each other; event 8 has 5, 6 and 7 as sub-events, also partially overlapping.

A narration n is a triple $\mathsf{n} = \langle s, k, \sigma \rangle$ consisting of:

1. A text $s \in S^\star$ giving the *content* of the narration, of length $|s|$.
2. A positive integer k giving the *depth* of the narration, that is the maximum number of *levels* in which the narration is structured. For instance, a narration structured in books and chapters has depth 3: level 1 is the level of the entire narration, level 2 is the level of books, and level 3 is the level of chapters. A narration that has no structure has depth 1. Note that depth is defined as a maximum, in order to capture the idea that not all levels need to be populated, e.g. not all chapters need to have sections: it is sufficient that one chapter has a section to have depth 3.
3. A function σ giving the *structure* of the narration. σ has the first k positive integers $\{1, 2, \ldots, k\}$ as domain and sets of intervals in $[1, |s|]$ as range. Each interval $[i, j]$ in the range of σ is called a *structural unit*, or simply unit, and its *content* is the sub-string of s from the i-th to the j-th character. $\sigma(1)$ is always the set containing

only the unit $[1, |s|]$, since the first level is the level of the entire narration. For $2 \leq j \leq k$, $\sigma(j)$ is a set of pairwise disjoint intervals, each one contained in one interval i of the previous level $\sigma(j-1)$ and giving the subdivision of i at the level j. Figure 4.1 right gives an example of the structure of a narration consisting of a text s of two thousand characters ($|s| = 2000$), divided in two chapters, one of fifteen hundred characters, the other of five hundred characters. Each chapter has one section partially covering its content.

This model of narration is kept simple to illustrate the concept for narrations with an acyclic structure, such as books. The model is not adequate to deal with narrations with possibly cyclic structures such as hypertexts. However, this is no real limitation, as it is always possible to capture arbitrary structures using more sophisticated models. Indeed, all the structures that can be used in a narration are expression of some grammar, therefore they can always be captured by a formal structure defined in set-theoretic terms.

Finally, the reference function was modelled. Given a fabula f and a narration n, a *reference function between* f and n, $\mathsf{ref}(\mathsf{f}, \mathsf{n})$, is a pair (F_{n}, r) where:

- F_{n}, the *fragmentation* of ref, is a set of intervals called *fragments*, each of which is contained in a unit of n, called the *source* of the fragment. Each fragment identifies the portion of the narration that narrates an event of the fabula, and has as content the sub-string of the source's content delimited by the fragment.
- r, the *event association* of ref, is an injective function assigning to each fragment f in F_{n} an event $e = r(f)$ that is one of the events of the fabula f, that is $r(f) \in E$.

The above definition is meant to leave maximum freedom in constructing the plot of the narration. In particular:

- Fragments can be derived from any unit of the narration, not only from those that belong to the highest level.
- Fragments can be freely chosen, allowing them to arbitrarily overlap. Therefore the injectivity of the event association, which imposes that two fragments may not narrate the same event, does not represent a limitation to the creativity of narrators: a piece of text may narrate two or more events simultaneously.
- Similarly, it was not imposed that the event association was surjective, so that each event in the fabula is associated to some fragment of the narration, leaving to narrators the possibility of omitting the narration of some events.
- Finally, it was not imposed that the narration of a sub-event was a part (technically, a sub-string) of the narration of the super-event. This condition may well apply to history texts, in which, e.g., the narration of the battle of Ludford is part of the narration of the War of the Roses. But it does not necessarily apply to other kinds of narrations, therefore it is not included in the model.

In this way, the plot of the narration can be displayed on a line, similarly to the fabula, except that in the fabula the line represents the flow of time, while in the narration it represents the sequence of characters that constitutes the content of the narration. Figure 4.2 illustrates this similarity between fabula and plot. The left-hand side of the Figure gives the reference function between the fabula and the narration presented in the previous examples (see Figure 4.1). The right-hand side shows the narration content against which both the plot (bottom) and the structure of the narration (top) are displayed.

Fragment f	$r(f)$
$[1, 500]$	4
$[600, 900]$	3
$[600, 800]$	1
$[820, 890]$	2
$[1250, 1850]$	8
$[1250, 1440]$	5
$[1501, 1700]$	6
$[1710, 1870]$	7
$[1950, 2000]$	9

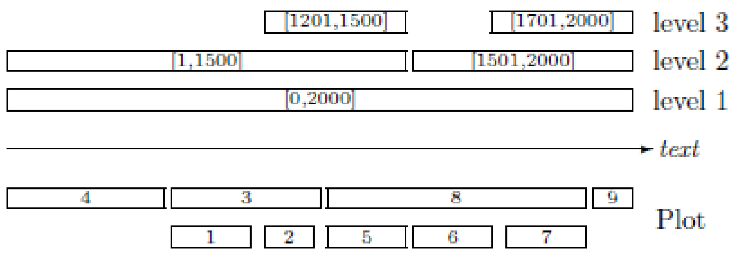


Figure 4.2: *A reference function (left) and the resulting correspondence between narration structure (top right) and plot (bottom right)*

Tying things up, a narrative is defined N as a $(k+1)$-tuple, $k \geq 1$,

$$N = \langle \mathsf{f}, (\mathsf{n}_1, \mathsf{ref}_1), \ldots, (\mathsf{n}_k, \mathsf{ref}_k) \rangle$$

where f is a fabula, and each pair $(\mathsf{n}_i, \mathsf{ref}_i)$, $1 \leq i \leq k$, consists of a narration n_i and a reference function between the fabula f and the narration n_i. This definition directly reflects the concept of narrative as spelled out in the conceptualization, that is as a fabula endowed with one or more narrations, each related to the fabula by a reference function.

CHAPTER 5

The Ontology Validation Experiment

This chapter presents the experiment to validate the conceptualisation presented in the previous Chapter. The experiment consists in using the conceptualisation for expressing a narrative based on a short biography of the Italian poet Dante Alighieri. Biographies are a very common kind of narratives in digital libraries: they provide an ideal context for the interpretation of the works included in the digital library, by relating these works with the events of the life of the person who created them. Dante Alighieri was chosen because his life is the subject of many studies, and we had the possibility of working in close cooperation with one authoritative historian of the University of Pisa, who is writing a revised version of Dante's biography, using the ultimate discoveries in this field. Cooperating with the historian allows us to directly access the primary sources of the knowledge represented by the biography, and also adds the possibility of having our representation of Dante's life evaluated by a true expert. The evaluation is presented in Chapter 7.

In order to carry out the experiment, two tools were used:

- an ontology to express the conceptualisation;
- a computer system to create, store and access the result of the experiment.

Concerning the first point, the conceptualisation was expressed using existing ontologies, with a preference for standards, for obvious interoperability reasons. Of course, existing ontologies have to be extended with notions that are suited to describe narratives. However, it was paramount to minimise the number of such extensions, in order to minimise the idiosyncrasies in this thesis.

For what regards the second point, a similar approach was followed, by maximising the re-use of software in order to minimise the effort in software development. Indeed, a well-know tool suite was re-used for the persistence and the access of the ontology and the semantic network for Dante's life. Only a tool for building narratives was therefore developed, since at the moment no other tool exists that can be adapted to needs of this research. Because this tool represents one of the main outcomes of this thesis, it is described separately in the next Chapter.

The present Chapter is structured as follows:

- Section 5.1 presents the selection of an existing ontology for expressing the conceptualisation, based on an analysis of two top ontologies, the CIDOC CRM (CRM for short) and DOLCE+ ontologies;
- Section 5.2 presents the expression of the conceptualisation in the selected ontology;
- Section 5.3 finally presents the narrative of Dante's life that was built on top of the ontology, using the tool presented in the next Chapter.

5.1 Selection of an Existing Ontology

Two top ontologies were analysed in order to understand whether their vocabularies are rich enough to capture the logic definitions of the components of narratives described in Chapter 4 (Section 5.1). The first analysed standard ontology is the CRM, a high-level ontology that allows to integrate the information contained in data of the cultural heritage domain along with their correlation with knowledge stored in libraries and archives [33]. The second analysed ontology is DOLCE-Lite-Plus (an extension of DOLCE), the first module of the WonderWeb Foundational Ontologies Library [86]. DOLCE aims at representing the ontological categories underlying natural language and human common-sense. DOLCE is described as an "ontology of particulars" [86], which the authors explain as an ontology of instances rather than an ontology of universals or properties. Particulars are entities that cannot have instances; universals are entities that can have instances. In linguistics, proper nouns are normally considered to refer to particulars, while common nouns to universals. DOLCE+ is an extension of DOLCE containing some modules dedicated to core ontologies of contexts, time, space, plans etc. The current implementation of DOLCE+ is DOLCE+ light-plus. DOLCE-Lite-Plus (shortly DOLCE+) has been used to align about 900 synsets from the noun hierarchies of the WordNet 1.6 English lexical database. WordNet is a "de facto" standard for many Natural Language Processing and Semantic Web applications. This alignment makes it possible to put the entire 66,000 synsets from WordNet 1.6(actually about 60,000 classes and 5,000 individuals) under DOLCE-LitePlus. The mapping between logic concepts and the definitions of the terms included in the CRM and DOLCE+ ontologies is important to evaluate if classes and properties defined in the two ontologies can be reused in the ontology for narratives. This mapping is reported below. The definitions and examples are extracted from both CRM and DOLCE+ official documentations.

- *Event*
  - The class "Event", along with its subclass "Activity", corresponds to the definition of Event in Event Calculus. In the CRM this class "comprises changes of

states in cultural, social or physical systems, regardless of scale, brought about by a series or group of coherent physical, cultural, technological or legal phenomena."
- In DOLCE+, "Event" is a subclass of "Perdurants". "Classically, endurants (also called continuants) are characterized as entities that are in time', wholly present (all their proper parts are present) at any time of their existence. On the other hand, perdurants (also called occurrents) are entities that happen in time, which extend in time by accumulating different "temporal parts", so that, at any time t at which they exist, only their temporal parts at t are present. Events are called achievements if they are atomic, otherwise they are accomplishments".

- *Action*
  - Actions identified by Davidson and defined in Situation Calculus correspond to the class "Activity" in the CRM. In CIDOC-CRM "this class comprises actions intentionally carried out by an actor that result in changes of state in the cultural, social, or physical systems documented. This notion includes complex, composite and long-lasting actions such as the building of a settlement or a war, as well as simple, short-lived actions such as the opening of a door."
  - In DOLCE+ Action as defined as "Perdurant that exemplifies the intentionality of an agent. Could it be aborted, incomplete, mislead, while remaining a (potential) accomplishment ... The point here is that having a result depends on a method, and then an action remains an action under incomplete results. As a matter of fact, if we neutralize intentionality, a purely topological, post-hoc view is at odds with the notion of incomplete accomplishments".

- *Generalized Events*
  - Generalized Events are represented in the CRM with the class "Period". "This class comprises sets of coherent phenomena or cultural manifestations bounded in time and space. It is the social or physical coherence of these phenomena that identify an E4 Period and not the associated spatio-temporal bounds. These bounds are a mere approximation of the actual process of growth, spread and retreat".
  - In DOLCE+ the class "Spatio-temporal-particular" corresponds to Generalized Event. In this class are those entities that are in space-time. The components of the types of generalized events were identified and tried to map with classes and properties of both ontologies.

- *Location*
  - This concept is represented in the CRM through the class "Place". "This class comprises extents in space, in particular on the surface of the earth, in the pure sense of physics: independent from temporal phenomena and matter".
  - In DOLCE+ "Space-region" class represents an ordinary space: geographical, cosmological, anatomical, topographic, etc.

- *Time*
  - This concept is represented in the CRM using the "Time-Span" class. "This class comprises abstract temporal extents, in the sense of Galilean physics, having a beginning, an end and a duration. Time Span has no other semantic connotations.

Time-Spans are used to define the temporal extent of instances of E4 Period, E5 Event and any other phenomena valid for a certain time".
- In DOLCE+ the concept of time is represented using the class "Temporal Region".

- *Start*
  - The concept of start is represented in the CRM with the property "starts". "This property allows the starting point for a Temporal Entity to be situated by reference to the starting point of another temporal entity of longer duration. This property is only necessary if the time span is unknown (otherwise the relationship can be calculated). This property is the same as the starts/started-by relationships of Allen's temporal logic" [4].
  - In DOLCE+ this concept is represented using the property starts that identifies "a beginning part of a perdurant (any part that includes the 'initial' boundary, but not the 'final' one)".

- *End*
  - This concept is represented in the CRM using the property "finishes". This property "allows the ending point for a temporal Entity to be situated by reference to the ending point of another temporal entity of longer duration".
  - In DOLCE + is used the property "concludes that" denotes "a last part of a perdurant (any part that includes the 'final' boundary, but not the 'initial' one)".

- *Duration*
  - The concept of duration can be mapped with the property "occurs during" in the CRM. This property "allows the entire Time-Span of a Temporal Entity to be situated within the Time-Span of another temporal entity that starts before and ends after the included temporal entity. This property is only necessary if the time span is unknown (otherwise the relationship can be calculated). This property is the same as the during/includes relationships of Allen's temporal logic" [4].
  - The temporally-includes represents the concept duration in DOLCE+. The property means that "all temporal locations of perdurant x are also temporal locations of perdurant y".

- *Before and After*
  - The properties "occurs before" and "occurs after" identify "the relative chronological sequence of two temporal entities. It implies that a temporal gap exists between the end of A and the start of B. This property is the same as the before/after relationships of Allen's temporal logic" [4].
  - In DOLCE+ the property precedes identifies a "temporal precedence between two perdurants. No further dependence is implied (e.g. mereological, causal)."

- *overlap*
  - In CIDOC-CRM the property "overlaps in time with" identifies "an overlap between the instances of E52 Time-Span of two instances of Temporal Entity. It implies a temporal order between the two entities: if A overlaps in time B, then A must start before B, and B must end after A. This property is only necessary if the relevant time spans are unknown (otherwise the relationship can be calculated). This property is the same as the overlaps/overlapped-by relationships of Allen's

temporal logic (Allen, 1983, pp. 832-843)." - DOLCE+ represents this concept using the property "temporally-overlaps" that "identifies a (partly) common temporal location".

- *Physical Objects*
  - In the CRM the class Physical Thing describes "all persistent physical items with a relatively stable form, man-made or natural. Depending on the existence of natural boundaries of such things, the CRM distinguishes the instances of Physical Object from instances of Physical Feature, such as holes, rivers, pieces of land etc."
  - In DOLCE+ physical objects are represented using the class "Non-agentive physical objects" that collects "objects like houses, body organs, pieces of wood, etc."

- *Mental Objects*
  - In the CRM the class "Conceptual Object" comprises "non-material products of our minds and other human produced data that have become objects of a discourse about their identity, circumstances of creation or historical implication. The production of such information may have been supported by the use of technical devices such as cameras or computers".
  - DOLCE+ defines a "Mental Object" as a percept, a sense datum. A private experience is an example of a mental object.

5.1.1 Discussion

The above analysis shows that both DOLCE+ and CRM are adequate to express the conceptualization of narratives. In particular, both possess the fundamental notions of the conceptualization, and the conditions mathematically stated on these notions in the conceptualization, can be implemented by creating suitable specializations of the corresponding notions. This choice falls on the CRM for the following reasons:

- The CRM is an ISO standard since 2006 (ISO21127:2006) and renewed 2014 (ISO21127:2014). As such, it offers a stronger guarantee under many aspects: it is widely known, it is regularly revised, it is universally accessible; at present, the same can be said also about DOLCE+, but DOLCE+ not being an ISO standard, it is unpredictable how long the present status will last;

- The CRM is specifically thought for the cultural heritage domain, and as such it is closer to the domain of narratives than DOLCE+, which is built with software engineering in mind. In fact, the CRM has been harmonized with the FRBR ontology (named as FRBRoo [34]), a core ontology for bibliographic information, and therefore it provides fundamental notions to model text, e.g. expressions and expression fragments, which are important for the aims of this thesis. Indeed, in many cases a narration represents a description expressed through a text and the ontology should be able to represent also its textual component;

- The Special Interest Group of the CRM continuously works for expanding the domain of applicability of the ontology, and a number of extensions have been already devised. One of these extensions, the CRMInf, was exploited during the evaluation of the ontology[1];

- A datalog-based, efficiently implementable logical specification of the CRM has been recently developed [94], which provides a basis for a practical experimentation of the ideas presented in this thesis.

For these reasons, the conceptualization presented above will be expressed in the CRM.

5.2 Specification of the Conceptualisation

In this Section the previously formalized conceptualization is expressed using the CRM. The core notions from the ontology will be used, with some additional classes and properties belonging to the CRMinf, an extension of the CRM for capturing inferential provenance.

Technically, the full OWL representation of the CRM, called Erlangen CRM/OWL[2] (ECRM for brevity), is used. ECRM is an OWL-DL 1.0 implementation of the CRM [53], and it can be defined as an interpretation of the CRM in a logical framework attempting to be as close as possible to the text of the specification[3]

In order to represent all the formal conditions (such as asymmetry and reflexivity) identified in the conceptualization, new CRM sub-properties are added to the official ones. Since OWL 1.0 only supports symmetric and transitive properties, the ECRM version was enriched with other conditions, such as reflexive, irreflexive and asymmetric properties, which are supported by OWL 2.0[4]. In the following, the sub properties that were introduced are reported and described in detail.

Events are expressed in the CRM as instances of class *E5 Event* (as a notational convention, the terms from the CRM vocabulary are written in italics) which "comprises changes of states in cultural, social or physical systems, regardless of scale, brought about by a series or group of coherent physical, cultural, technological or legal phenomena"[5]. In fact, in the CRM it is possible to also express actions, as instances of class *E7 Activity*, which comprises "actions intentionally carried out by instances of *E39 Actor* that result in changes of state in the cultural, social, or physical systems documented". Compliant to the view of this thesis, class *E7 Activity* is a sub-class of *E5 Event* which in turn is a sub-class of *E4 Period.*

Time intervals are represented in the CRM by the instances of class *E52 Time-Span*, which "comprises abstract temporal extents, in the sense of Galilean physics, having a beginning, an end and a duration".

Texts are represented in the CRM by the instances of the class *E62 String*.

Now a representative for fabulae has to find. Although the CRM is not designed to represent narratives, class *E28 Conceptual Object* generally fits the conceptualization

[1] For more information about the CRM extensions, see `http://new.cidoc-crm.org/collaborations`

[2] `http://erlangen-crm.org/`

[3] The ECRM implementation was developed by Bernhard Schiemann, Martin Oischinger and Günther Görz at the Friedrich-Alexander-University of Erlangen-Nuremberg, Department of Computer Science, Chair of Computer Science 8 (Artificial Intelligence) in cooperation with the Department of Museum Informatics at the Germanisches Nationalmuseum Nuremberg and the Department of Biodiversity Informatics at the Zoologisches Forschungsmuseum Alexander Koenig Bonn.

[4] `https://www.w3.org/TR/owl2-overview/`

[5] All quotations in this Section are from the CRM specification, available from `http://www.cidoc-crm.org/docs/cidoc_crm_version_6.2.pdf`

of fabula, since it "comprises non-material products of our minds and other human produced data that have become objects of a discourse about their identity, circumstances of creation or historical implication".

Next, a CRM property that links the events in a fabula with the fabula itself was found. CRM property *P12 occurred in the presence of (was present at)* is the closest approximation that the CRM currently offers for this purpose. The domain of this property is class *E5 Event* and its range is class *E77 Persistent Item,* a super-class of *E28 Conceptual Object.* The inverse of this property "describes the active or passive presence of an *E77 Persistent Item* in an *E5 Event* without implying any specific role". As such, the property is used to connect a thing to the event in which the thing was present. By interpreting fabulae as things, the property can be used also for the purposes of this thesis.

The relations defined on the events of the *fabula* are expressed by the following CRM properties:

- the *direct part-hood* relation is represented by a newly introduced property *narr direct part,* defined to be a sub-property of the CRM property *P9 consists of (forms part of)* which "associates an instance of *E4 Period* with another instance of *E4 Period* that is defined by a subset of the phenomena that define the former". That direct part-hood is a partial, acyclic function over events; therefore in OWL *narr direct part* is defined to be a functional object property. In order to capture its acyclicity, a transitive super-property of *narr direct part*, called *narr acyclic part*, is introduced and it is imposed that the latter property be an irreflexive property. In this way, every cycle introduced on *narr direct part* will make the knowledge base inconsistent. On the other hand, since the CRM property *P9* is defined as transitive, it was extended to be reflexive thereby modelling indirect part-hood.
- the *event beginning* and the *event ending* functions are at once represented by the CRM property *P4 has time-span (is time-span of)*, which "describes the temporal confinement of an instance of an *E2 Temporal Entity*" and therefore of an event. In this way, the condition that the beginning of an event always precedes the ending of the same event is an automatic consequence, and these points can be recovered through the property *E49 Time Appellation.* Because the period of occurrence of an event may not be known, the CRM allows to directly relate events based on their occurrence time. To this end, it introduces seven properties (*P114* to *P120*) mirroring the temporal relations formalized by Allen's temporal logic [5]. For instance, the CRM property *P117 occurs during (includes)* "allows the entire time-span of a temporal entity (including events) to be situated within the time-span of another temporal entity that starts before and ends after the included temporal entity", and mirrors the *during* relationship of Allen's temporal logic. To be compliant with the conceptualization, the sub property *narr occurs before* was added. Indeed, the CRM defines this property as transitive, but in the conceptualization is reported as irreflexive and asymmetric. For this reason, the classes owl:IrreflexiveProperty and owl:AsymmetricProperty were assigned to *narr occurs before.*
- the *causality* relation is represented by introducing a new property of *causal dependency*. This is the only proposed extension to the CRM, whose only causal property *P17 was motivated by* cannot be used for narratives since it relates activ-

ities but not events. Indeed, CRMsci [6], an extension of CRM for science, defines a direct causality relation, via the property *O13 triggers,* which "associates an instance of *E5 Event* that triggers another instance of *E5 Event* with the latter (...); in that sense it is interpreted as the cause". However, this property is inadequate to the needs of narratives, whose events may be separated by possibly long periods of time. The *causal dependency* property was defined as transitive and reflexive.

Turning to the expression of narrations, narrators are represented by the instances of the CRM class *E21 Person*. The authoring relation between a narrator and the text of the narration is represented through the event of the creation of the text, an instance of the CRM class *E65 Creation*. This creation event connects to the narrator by the CRM property *P14 carried out by* and to the narration by the CRM property *P94 has created*. The narration itself is an instance of the CRM class *E73 Information Object*, which identifies "immaterial items, such as poems, data sets, images, texts, multimedia objects [...] that have an objectively recognizable structure and are documented as single units."

The mereology of the text is represented using the CRM property *P106 is composed of*, connecting a structural whole to its parts. However, *P106* represents the structure of the text as defined by the author, and the units of this structure not necessarily coincide with the units that narrate a single event. It may happen that a structural unit narrates more than one event, but even if it narrates a single event, the event may be narrated in a small portion, or fragment, of a unit (e.g., the portion "Dante was baptised in Florence" of the sentence "On Saturday 26 March 1266 Dante was baptised in Florence, as he states himself in Inferno XIX 17"). In order to factor out the chunks of text that narrate a single event, therefore a specific class is used, the FRBRoo class *F23 Expression Fragment*. FRBRoo [34] is a bibliographic ontology resulting from the harmonization of the FRBR ontology and the CRM. As such, it extends the CRM with bibliographic-specific classes and properties. In particular, class *F23* comprises "parts of Expressions and these parts are not Self-Contained Expressions themselves" and is a subclass of *E73 Information Object.* Expression fragments are connected to the structural units of text where they belong via the *P106* property.

Finally, in order to express the relation between a fragment of text and the event narrated by the fragment, the proposed idea is that a fragment is related to a number of propositions, which collectively formalize the content of the fragment. Such relation is expressed by the CRM property *P129 is about*. Each so related proposition is an instance of the CRM class *E73 Information Object* and is structured according to the Resource Description Framework (RDF) [85] as consisting (not exclusively) of a subject, a predicate and an object. The relations between a proposition and its constituent parts are expressed by borrowing three properties borrowed from the RDF vocabulary: *rdf:subject*, *rdf:predicate* and *rdf:object.* Overall, then, the relation between a fragment of text and the narrated event is obtained as follows: an expression fragment is about some proposition having the event as a subject (see Figure 5.3). This completes the ontology for narratives.

[6] http://www.ics.forth.gr/isl/CRMext/CRMsci/docs/CRMsci1.2.2.pdf

5.3 Experimentation: The Dante Alighieri Case Study

In order to validate the ontology in a realistic setting, the ontology was used to represent a narrative of a Dante Alighieri's biography as case study. To this end, a collaboration with an authoritative Dante's biographer who has provided the text from which the plot [64] was extracted, was established. The biographer was also interested in expressing the primary sources supporting the plot, so the ontology was enriched to also document the process through which a certain narrative came into existence. To this end, the construction of a narrative by an historian was viewed as an inferential process, using evidence collected from sources to infer propositions that were subsequently narrated in a text. The primitives for modelling inference are borrowed from another extension of the CRM, the CRMInf, an ontology for capturing argumentation and inference making in descriptive and empirical sciences. The CRMInf is still a proposal for approval by the Special Interest Group of the CIDOC CRM [7]. The usage of the CRMInf will be illustrated in due course, along with other minor aspects of the ontology, which helped in contextualizing the represented knowledge. Generally speaking, the use of the CRMInf allowed describing the knowledge *provenance*, as the process of tracing the origins of knowledge [27].

5.3.1 The Semantic Network

The semantic network to model Dante's biography and its provenance includes the *fabula* and the *narration*, as defined in the conceptualization; in addition, it also includes the *provenance*, which represents the inferential process of the narrator who composes a narrative from primary sources whose contents are propositions about the events in the fabula. In the following examples, CRM classes and properties are named using the CRM names, while terms re-used from other vocabularies are named by prefixed qnames, e.g. *rdf:type*. Class instances are named in lower-case strings ended by a number in Sans Serif type, e.g. e1.

Fabula Figure 5.1 shows a graphical view of part of the fabula, including just two events: e1, the birth, and e2, the baptism, of Dante Alighieri. Both these events are instances of *E5 Event*, in addition e1 is an instance of *E63 Beginning of Existence*, which is a sub-class of *E5 Event*. Note that the *rdf:type* property from the RDF vocabulary to model instance-of was used. Dante himself is represented by object d, an instance of *E21 Person*. The property *narr occurs before* links the event e1 with the event e2.

In addition to the fabula's properties, an event is contextualized in terms of other properties, which are used heavily in the validation experiment. These factual properties can be grouped as: *Where* and *When* an event happened, and *Who* (persons) and *What* (things) were involved in it [120, 130]. In particular:

Where is represented by property *P7 took place at*, which links an event with the instance of *E53 Place* giving the location of the event, or by property *P8 took place on or within (witnessed)*, in case the location of an event is described in terms of a physical object (such as the baptism of Dante taking place in the Baptistery of Florence).

When is represented by property *P4 has time-span*, which links an event with the instance of *E52 Time Span* giving the period of occurrence of the event.

[7] http://www.ics.forth.gr/isl/CRMext/CRMinf/docs/CRMinf-0.7.pdf

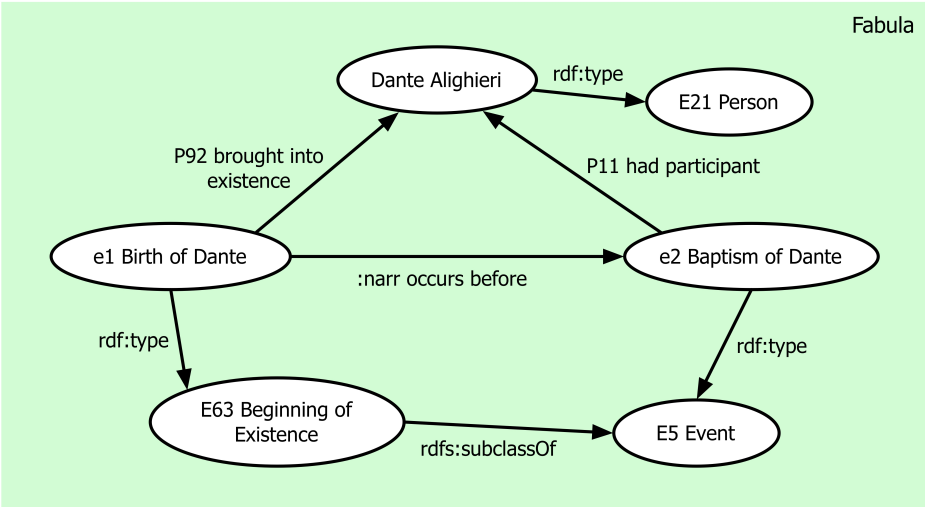


Figure 5.1: *A portion of the* fabula

Who is represented by property *P11 had participant*, which links an event to the instance(s) of *E39 Actor*, giving the person(s) or group of persons who participated to the event. If the event is an action, property *P14 carried out by* is used as predicate in the proposition to link the action to the corresponding actor. In order to specify the nature of the actor's participation in an event, a reification is required for the representation of the role in RDF/XML format. An event is related to an entity that represents the actor and the corresponding role. This entity is an instance of the class *ActorWithRole*, which was introduce as a new class of the ontology. The event is related to the corresponding instance of the *ActorWithRole* using the new property *had participant* that was created for this aim. This entity has two properties: (i) *role* that links it to a string that defines the role of the actor in natural language and (ii) *subject* that links the entity to the instance of the class *E39 Actor*, who is the person or the person group who participates to the event.

What is represented by property *P12 occurred in the presence of/ was present at*, which relates an instance of *E5 Event* with an instance of *E77 Persistent Item*. For example, the baptism of Dante *P21 occurred in the presence of* the baptismal font, which is an instance of *E19 Physical Object*. *P12* can also be applied to concepts, which are represented as instances of *E89 Propositional Object*. For example, "Rhetoric" as mentioned in "Dante studied rhetoric with Brunetto Latini". Finally, an event was related with: (a) a digital object that can be used as a representation of it (e.g. the engraving "The Death of Beatrice"[8]), through the property *P138 represents*; and (b) to a textual annotation (e.g. a comment in natural language that the narrator can write in order to add some additional explanation to the event) through the property *P3 has note*.

Narration Figure 5.2 shows a portion of the *narration* of the life of Dante Alighieri. At the bottom of the Figure, the biography is represented by object `io1`, an instance of the class *E73 Information Object,* resulting from the creation event `c1`, represented as an

[8] http://www.europeana.eu/portal/record/9200365/BibliographicResource_1000055750062.html

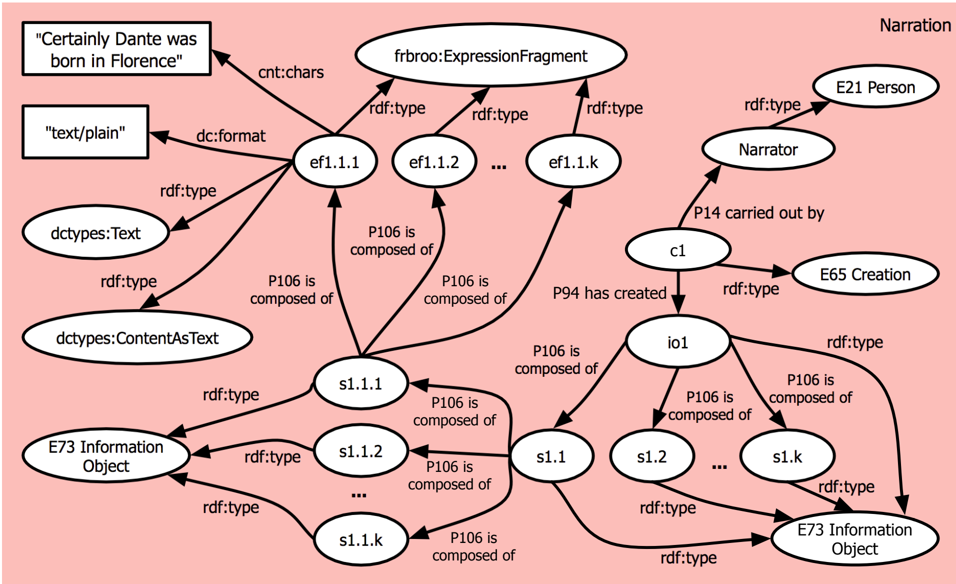


Figure 5.2: *The* narration.

instance of the class *E65 Creation*, carried out by (*P14 carried out by*) the biographer, who is an instance of the class E21 Person. Although the following information was not reported in Figure 5.2 for lack of space, in order to assign a label to the biographer that represents his/her name in natural language, the property *P47 is identified by* was used. This property relates the instance of the *E21 Person* class representing the biographer with an instance of the class *E41 Appellation*. Then, using the property *P3 has note* the instance of the class *E41 Appellation* is related with a string that represents the name of the biographer. At the top of the figure, the biography is structured in chapters, paragraphs and textual fragments. The class *E73 Information Object* was used for representing chapters and paragraphs and *Expression Fragment* of FRBRoo [34] for textual fragments.

Each instance of *Expression Fragment* has a textual content which is modelled according to the recommendations of the W3C's Content in RDF[9], as shown in the top left corner of Figure 5.2. Specifically:

1. the property *format* of the Dublin Core[10] giving the MIME media type of the instance. This allows distinguishing, e.g. embedded content in plain text from content encoded in HTML;
2. the property *chars* in the Content Namespace in RDF, giving the sequence of the characters of the content;
3. *dctypes:Text* class, to indicate that the instance represents a resource primarily intended to be read;
4. *cnt:ContentAsText* class, to indicate that the instance represents textual content.

Provenance Figure5.3 shows a portion of the provenance, centred around proposition `p1`. This proposition expresses that its subject, the birth of Dante (event `e1`), stands

[9] `http://www.w3.org/TR/Content-in-RDF10/`
[10] `http://dublincore.org/`

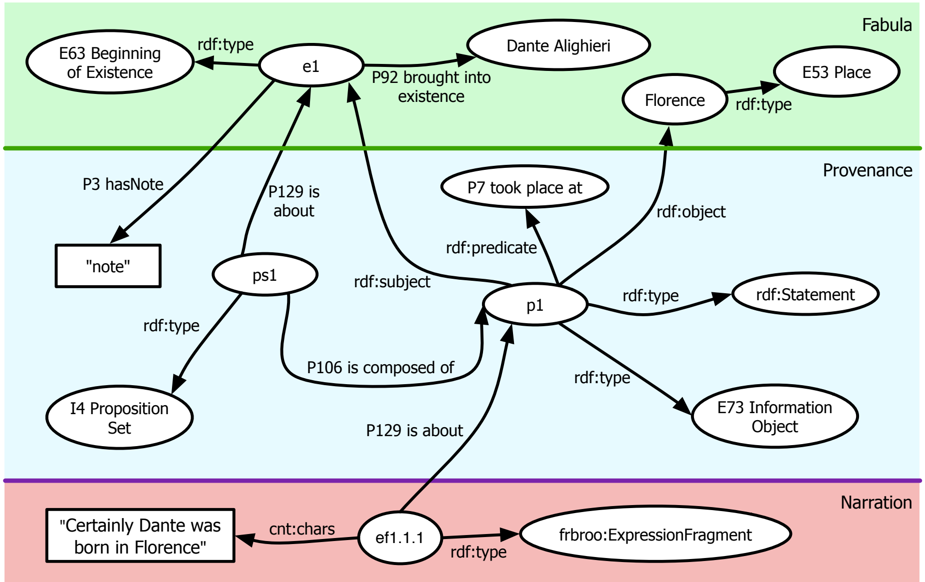


Figure 5.3: *The structure of the proposition p1.*

according to its predicate *P7 took place at* with its object, Florence. Therefore, p1 says that Dante was born in Florence. Note that p1 is part of ps1 that is a *I4 Proposition Set*. As it will be shown below, proposition sets play a prominent role in the CRMInf.

Figure 5.4 shows another portion of the *provenance*, representing the inferential process of the biographer. Everything starts from the reading of a primary source, modelled as an event of observation, o1, that is an instance of the class *S4 Observation*. o1 *O8 observed* the observable entity oe1, which is an instance of the class *S15 Observable Entity*, while o1's *O16 Observed value* is a set of propositions ps2, represented with the class *I4 Proposition Set*. Because they result from direct observation, the propositions in ps2 are believed to be true by the biographer, so ps2 *J4 is subject of* a belief b1, represented with the class *I2 Belief*, which *J1 was the premise for* an instance of the class *I5 Inference Making* im1, which *J2 concluded that* belief b2 is the case. b2 is the belief of a set of propositions ps1 which, as it has already been mentioned, includes the proposition p1 asserting that Dante was born in Florence. So, the connection between the reading of the Divine Comedy and the proposition that Dante was born in Florence is established through an inference making event, between the propositions resulting from the reading and the propositions including the conclusion. Inference making, in turn, requires that inference maker believes both the premises and the conclusions, so beliefs are involved in the process. In order to represent the textual fragment of the primary sources and the corresponding bibliographic references, two new properties were added. The property *has textual fragment* relates the proposition with an instance of the class *Expression Fragment* and represents the textual fragment of the primary source to which the proposition is related. The property *has reference* relates the instance of the class *Expression Fragment* with a string that represents its bibliographic references.

In order to assign a label representing the name of the observed primary source in natural language, the property *P47 is identified by* was used to relate the instance of the class *S15 Observable Entity* representing the source with an instance of the class *E41 Appellation*. Then, using the *P3 has note* property, the instance of the class *E41*

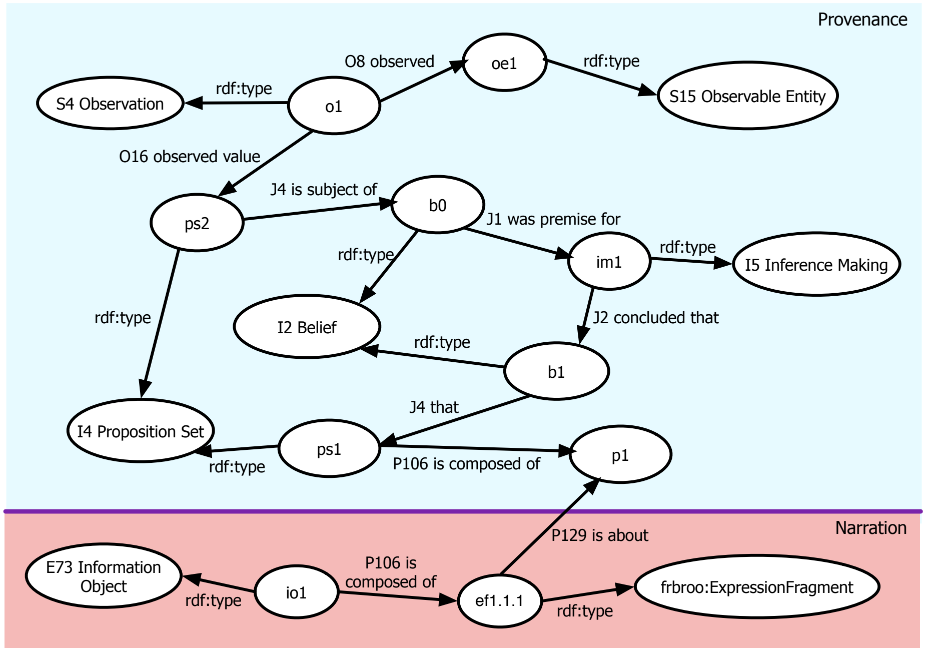


Figure 5.4: *Representation of the* provenance.

Appellation is related with a string that represents the title of the primary source. In order to facilitate the extraction of the knowledge from the knowledge base, a direct property that relates the proposition with the corresponding primary source was added. This property was introduced as new property and was called *has source*.

Some Data about the Semantic Network

In order to make the experiment of creating a narrative of the Dante Alighieri's life and works, a portion of the biography of the poet written by an expert of Dante's [64] was taken into account. The document produced by the scholar was MS Word document of 40 pages (83,688 characters), which summarized his complete work. The text was manually analysed in order to extract the events contained. Then, using a semi-automatic tool developed for this research, the corresponding narration was created. In about five hours of work a semantic network composed of 81 events linked by causal, temporal and mereological relations, and 206 propositions was created. This was possible because the tool provides a significant support to build the fabula and the narration (as described in detail in Chapter 6), while requiring from the user only the minimal information to build the provenance of knowledge. In fact, the tool automatically automatically constructs the objects, such as proposition sets, beliefs and inference making objects that are part of the network but do not bring *per se* primary semantic information content. Once the user has completed the creation of the narrative, the events, their entities and relations are saved to a JSON object. A Java software was developed to automatically translate the JSON object to an OWL (Web Ontology Language) graph, in which the knowledge is represented as triples subject-predicate-object. The resulted graph is composed of 7,379 triples. In order to provide a description of the created semantic network, some data are reported below. Figure 5.5 reports data about the classes

Classes used in the semantic network

CLASS NAME	INSTANCES
I2 Belief	412
E73 Information Object	252
I4 Proposition Set	206
I5 Inference Making	206
E41 Appellation	173
E5 Event	81
E52 Time-Span	80
F23 Expression Fragment	78
ActorWithRole	40
E39 Actor	35
S4 Observation	29
E53 Place	28
E89 Propositional Object	4
E19 Physical Object	2

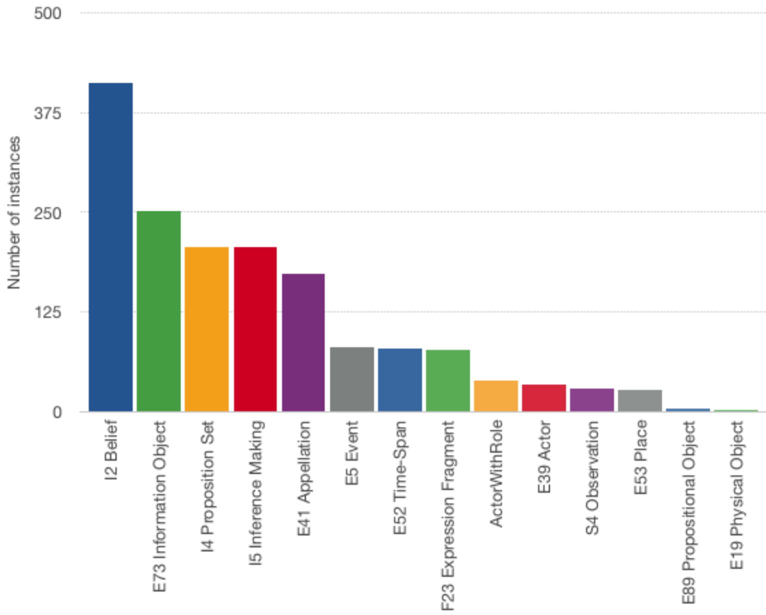


Figure 5.5: *Classes used in the semantic network.*

used in the semantic network. The data reported for the class *Information Object* in the Figure are the sum of the numbers of the propositions (206), the literary works (38), and the digital objects (8), since all these entities are instances of the class *Information Object*.

In Figure 5.6 data about the properties used to link the entities with the related events are reported. Figure 5.7 shows the data regarding the properties that relate the events in the narratives.

Figure 5.8 shows the data about the classes of resources that contextualize the events that compose the narrative. In this case, the data about the the class *Information Object* is limited to the literary works directly related to the events (and not those related to the propositions).

Figure 5.9 reports the list of types of events used in the narration.

Figure 5.10 reports the data about the entities re-used from Wikidata and those created by the user. It is interesting to note that about 2/3 of the entities are imported from Wikidata and only 1/3 was inserted manually. Figure 5.11 shows the comparison of the usage in the narration of the Wikidata entities and of the entities created by the users. The Wikidata entities are used in 84% of the prepositions related to the events that compose the narrative.

Properties used in the semantic network

PROPERTY NAME	INSTANCES
P106 is composed of	287
J4 that	206
J2 concluded that	206
J1 was premise for	206
P3 has note	182
P149 is identified by	173
hadParticipant	114
role	114
subject	114
P129 is about	78
P4 has time-span	80
P79 beginning is qualified by	80
P80 end is qualified by	80
P7 took place at	71
numChapter	41
numBook	41
P12 occurred in the presence of	22
P138 represents	9

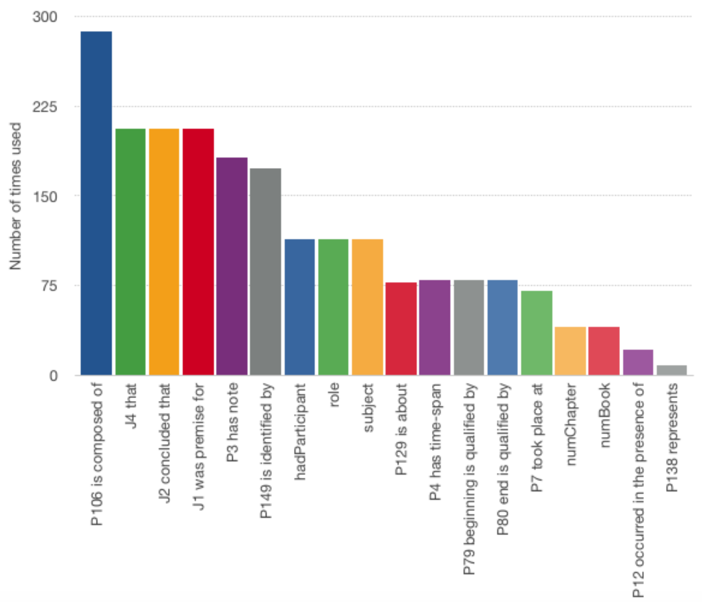


Figure 5.6: *Properties that link the entities with the related events.*

Temporal, causal and mereological relations

PROPERTY	INSTANCES
narr occurs before	79
narr consists of	15
causally depends on	14

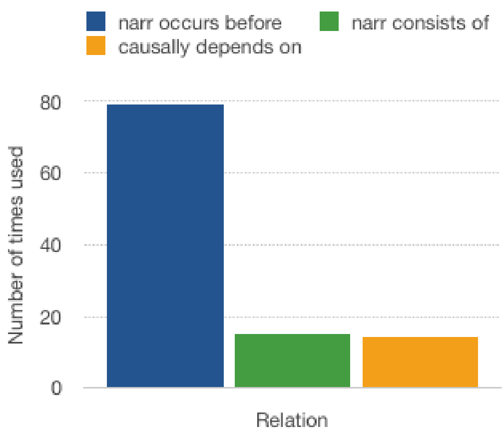


Figure 5.7: *Properties that relate the events in the narratives.*

Classes of resources related to events

CLASS	INSTANCES
E39 Actor	35
E53 Place	28
E73 Information Object	23
E89 Propositional Object	4
E19 Physical Object	2

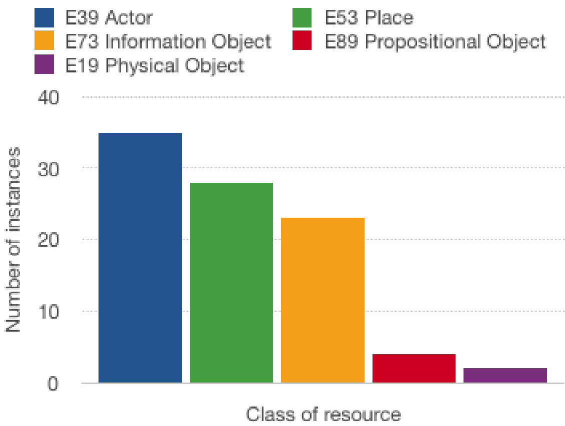


Figure 5.8: *The classes of resources related to events.*

Event types

EVENT TYPE	INSTANCES
Residency	16
Death	10
Election	8
Meeting	6
Travel	6
Creation	5
Battle	4
Marriage	4
Coronation	3
Education	3
Joining	3
Legation	2
Sentence	2
Birth	1
Baptism	1
Exile	1
Expedition	1
Lawmaking	1
Leaving	1
Lecture	1
War	1

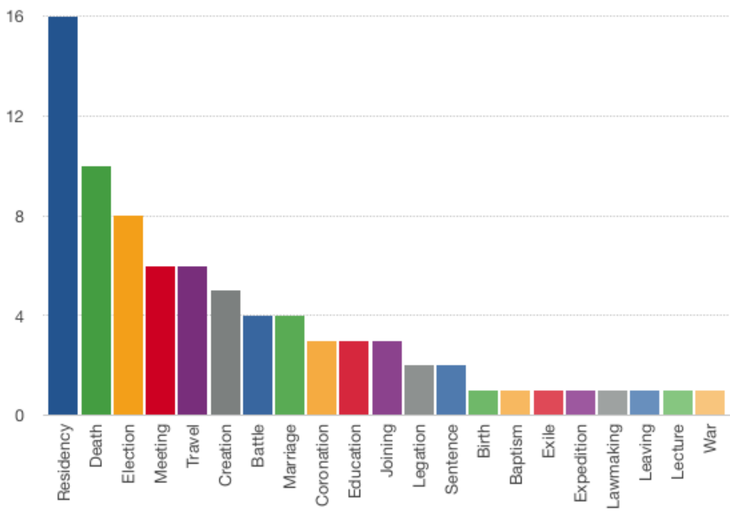


Figure 5.9: *Types of events used in the semantic network.*

Number of Wikidata vs. user-defined entities

PARTICIPANT	UNITS SOLD
Wikidata entities	61
User-defined entities	27

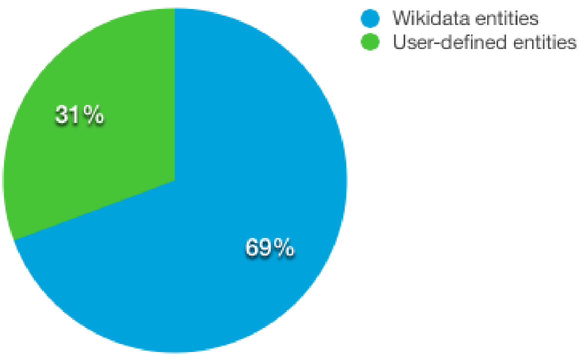


Figure 5.10: *Entities re-used from Wikidata and created by the user.*

Usage of Wikidata vs. user-defined entities

PARTICIPANT	UNITS SOLD
Wikidata entities	173
User-defined entities	33

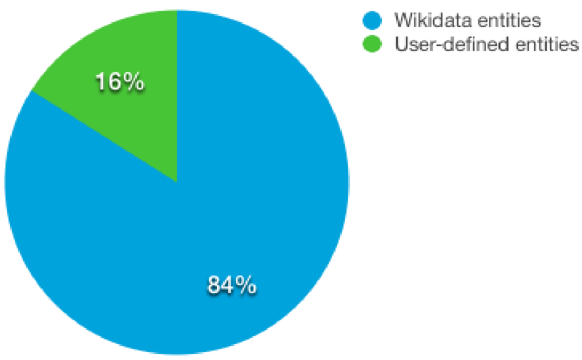


Figure 5.11: *Usage of the Wikidata entities and of the entities created by the users.*

CHAPTER 6

A Tool To Construct Narratives

An ontology without a validation experiment is an exercise in applied philosophy. In order validate the ontology in a realistic experiment and also to ease the adoption of the ontology, a tool for the creation of narratives based on the ontology was designed and implemented. The tool is presented in this Chapter. An overview of the related works is first given, followed by a detailed description of the tool.

6.1 Related Works

Several annotation tools for the semantic encoding of texts have been developed in order to create narratives. For example in [38] an annotation tool that allows users to create formal propositions to represent spans of text, as well as temporal relations and other aspects of narrative is presented. A built-in natural language generation component regenerates text from the formal structures, which eases the annotation process. The Story Workbench [44] is an annotation tool that provides a generic platform for text annotation. It allows performing common operations, including text processing (e.g. tokens, sentences, parts of speech extraction), functions (e.g. checking annotation validity by rule, manual manipulation of annotations) and tools (e.g. merging doubly-annotated texts into a single file). Semantator [123] is a semi-automatic tool for document annotation based on an ontology. After loading a text document and an ontology, Semantator supports the creation of ontology instances for any document fragment, by linking instances through the properties defined in the ontology. In [79] an annotation schema for the narrative features of media objects that relies on a formal theory of story and characters is described, along with a "Cinematic" tool for the annotation of video objects and automatic editing of the annotated objects. Narrative frameworks have been also created, as in [106] where the SONNET narrative framework is presented. This is a flexible framework to integrate, select, and re-use narrative models. The framework provides a lightweight ontology language for the definition of key terms and relationships among them. The framework specifies model metadata that allows developers to discover and understand models more readily. [129] presents a generic semi-automatic language-independent framework to annotate content using ontologies as a controlled vocabulary for annotations. Further, it uses Conceptual Graphs (CG) [125] as annotation language. The Dynamic Narrative Annotation Tool (DNA-t) allows creating semantic annotations in CG. After some words are selected in the annotated text, DNA-t can automatically create concepts as nodes of the graph. Each selected word appears as a newly created node. The user then finds an appropriate conceptual type in the ontology browser of DNA-t and assigns it to the newly created concept using drag-and-drop. The resulting annotation can be exported into different formats: OCML, RDF, DAML-OIL, and OWL. [48] describes some linguistic and statistical tools for historians and social historians in order to study events (such as strikes, demonstrations and other types of collective conflict). These tools are generally based on the semantic grammars approach. Semantic grammars organize information around the structure of natural language: noun phrase/verb phrase, or subject, action, object and their modifiers (e.g. time, space). Semantic grammars can be easily implemented using relational database systems (RDBMS). The data are suited for the application of new tools of data analysis such as network models. In [3] a project that develops an ontology for the domain of artists and paintings is presented. A selection of information extraction tools and techniques is developed and applied to automatically populate the ontology with information extracted from on-line documents based on the given ontology's representations and the WordNet lexicons. Then, narrative construction tools are developed to query the knowledge base through an ontology server, in order to search and retrieve relevant facts or textual paragraphs and generate a specific biography. The automatic generation of tailored biographies focusses on two domains. First, providing biographies for artists for whom some information exists on the Web; this may mean constructing text from basic factual information or combining text from a number of different sources. Second, the project aims at providing biographies that are tailored to the particular interests and requirements of a given reader. All these tools present pros and cons, but none of them can be directly applied to this project for the fundamental reason that the ontology is too specific, and adapting any one of these tools to work with it requires an effort comparable to implementing a tool from scratch that is tailored to the ontology.

6.2 The Tool: A Detailed Description

The tool to construct narratives, called *narratives building tool* (NBT for short), was designed for two kinds of user: (i) scholars who want to create a narrative starting from an already existing narration written by them, or (ii) general users (e.g. school teachers, students) who want to create a narrative based on a narration written by someone else or who want to represent a narrative that does not have a textual narration, e.g. the narration only exists in the user's mind. The tool was written in HTML5/CSS/JavaScript using the jQuery1, jQuery UI2, Bootstrap3, and Typeahead.js4 libraries. It uses an IndexedDB5 database to store data locally and retrieves data at subsequent loadings of the tool. TNBT uses the knowledge stored in the Wikidata knowledge base[1]. Wikidata, a project of the Wikimedia Foundation, is a free and open knowledge base that accepts editing by any user, following the model of Wikipedia. Wikidata currently contains more than 17 million items. Wikidata runs on the MediaWiki software[2] and the format it uses to store knowledge is document-based. However, it has recently reached full compatibility with Semantic Web technologies and provides a SPARQL endpoint (named Wikidata Query Service) to query the knowledge base. It is also fully integrated with Wikipedia. For these reasons, it was used as reference for the tool. In particular, NBT uses Wikidata to import a series of entities[3] related to the subject of the narration. These entities can then be connected to the events that compose the narrative and that are created by the user. There are two main ways to query Wikidata: the Wikidata API[4], which is the API provided by the MediaWiki software[5], and the Wikidata Query Service[6] (WQS). The WQS allows users to perform SPARQL queries on the whole knowledge base. The main interface of the tool is made up of three views, which correspond to the three phases of the narrative creation process:

1. The first view is used to select the subject of the narrative.
2. The second view allows creating the events that compose the fabula of the narrative.
3. The third view allows establishing the causal and mereological relations between the events of the fabula.

Before describing the interface of the tool in details, the mapping between the CRM classes and the Wikidata entities is reported in Section 6.2.1. This mapping is necessary in order to use the knowledge in Wikidata.

6.2.1 Wikidata–CRM Mapping

In order to re-use Wikidata entities, a mapping between Wikidata classes onto the CIDOC CRM classes adopted in the ontology was performed[7]:

- *Q5 Human // E21 Person*
- *Q16334295 Group of Humans // E74 Group*
- *Q223557 Physical Object // E19 Physical Object*
- *Q8205328 Artefact // E24 Physical Man-Made Thing*
- *Q7184903 Abstract Object // E89 Propositional Object*
- *Q17334923 Location // E53 Place*
- *Q386724 Work // E73 Information Object*

[1] `http://cacm.acm.org/magazines/2014/10/178785-wikidata/fulltext`
[2] `https://www.mediawiki.org/wiki/MediaWiki`
[3] Contrary to the standard RDF terminology, Wikidata generally refers to resources as *entities*, or more simply as *items*.
[4] urlhttps://www.wikidata.org/w/api.php
[5] `https://www.mediawiki.org/wiki/API:Main_page/it`
[6] `https://query.wikidata.org/`
[7] Wikidata classes are generally written lowercased, but for consistency with the CRM the title case throughout this section will be used.

- *Q234460 Text // E33 Linguistic Object*
- *Q107715 Time // E52 Time-Span*
- *Q1190554 Event // E5 Event*

Although this mapping seems sufficient to make the two ontologies compatible, some significant issues were observed:

- The first issue is that most of the entities in Wikidata are instances of more than one of the basic classes. For instance, a *Q5 Person* is also a *Q223557 Physical Object*, a building can be both a *Q223557 Physical Object* and a *Q17334923 Location*, and an *Q5061 Administrative Territorial Entity* is both a place and an organization. Thus, the mapping was not sufficient, but also a classification algorithm to sort the entities according to an interpretation priority had to be created. For instance, in Wikidata an administrative territorial entity (e.g. a city) is both a *Q17334923 Location* and a *Q16334295 Group of Humans*, but for the scopes of this thesis it had to be classified as a location. In fact, if the classification as a group of humans came before that of location, all the territorial entities would be wrongly classified according to the ontology.
- The second issue is that sometimes Wikidata and the CRM are misaligned. For instance, the CRM class *E21 Person* and the CRM class *E74 Group*, representing groups of people, are both subclasses of *E39 Actor*. In Wikidata, *Q16334295 Group of Humans* and *Q5 Human* are solely connected through the part-whole property named *P527 part of*. Therefore. if the two classes have to group together, they were selected separately from Wikidata.
- A third issue is that sometimes a Wikidata user inadvertently makes an edit that changes the structure of the ontology in a major way. This happened twice during the development of the tool: first when a user changed the definition of a subclass of *Q386724 Work*, and second when another user changed the definition of *Q515 City*. The first case affected a few literary works only, with relatively minor consequences. However, the second one indirectly made every city a subclass of event, which was a major mistake. In the first case the issue was solved by simply pointing it out to Wikidata users, in the second case the mistake was corrected by the author of the thesis who had to become Wikidata contributor and actively solve it in the source.

Summarizing, the open and collaborative nature of Wikidata can provide both advantages and disadvantages. However, despite these issues a mapping algorithm was developed. In Figure 6.1 the JavaScript function currently used to classify the entities is shown.

This function takes as input an array of Wikidata classes and returns the corresponding class in the ontology. In particular, the algorithm works as described in the following:

1. It initially excludes all entities with class *Q15474042 MediaWiki Page* from the mapping, as these represent pages from Wikipedia and other projects and are not needed for the purposes of this research.

```
function classFromArray(array) {
        if (array.indexOf('Q15474042') > -1)        return null;
        else if (array.indexOf('Q186081') > -1)     return 'E52';
        else if (array.indexOf('Q5') > -1)          return 'E21';
        else if (array.indexOf('Q234460') > -1)     return 'E73';
        else if (array.indexOf('Q8205328') > -1)    return 'E73';
        else if (array.indexOf('Q41176') > -1)      return 'E19';
        else if (array.indexOf('Q17334923') > -1)   return 'E53';
        else if (array.indexOf('Q16334295') > -1)   return 'E39';
        else if (array.indexOf('Q43229') > -1)      return 'E74';
        else if (array.indexOf('Q386724') > -1)     return 'E73';
        else if (array.indexOf('Q1190554') > -1)    return 'E5';
        else if (array.indexOf('Q7184903') > -1)    return 'E89';
        else if (array.indexOf('Q223557') > -1)     return 'E19';
        return null;
}
```

Figure 6.1: *The JavaScript function used to map Wikidata classes onto CIDOC CRM classes.*

2. It makes all entities with class *Q186081 Time* instances of the CRM class *E52 Time-Span*.
3. It makes all entities with class *Q15474042 Event* instances of the CRM class *E5 Event*.
4. It assigns all entities with class *Q5 Human* to the CRM class *E21 Person*.
5. It assigns all entities from classes *Q234460 Text* to the CRM class *E33 Linguistic Object*.
6. It makes all entities from classes *Q8205328 Artifact* and *Q41176 Building* instances of the CRM class *E24 Physical Man-Made Thing*.
7. It makes all entities from class *Q17334923 Location* instances of the CRM class *E53 Place* (buildings are excluded, territorial entities are included).
8. It assigns all entities from class *Q16334295 Group of Humans* to the CRM class *E74 Group* (territorial entities are excluded).
9. It assigns all entities from class *Q386724 Work* (that are not also artifacts or buildings) to the CRM class *E73 Information Object*.
10. It assigns all entities from class *Q7184903 Abstract Object* (that are not works or texts) to the CRM class *E89 Propositional Object*.
11. Finally, it makes all the remaining instances of *Q223557 Physical Object* subclasses of *E19 Physical Object*.

It should be noted that all the steps are executed *in this specific order,* otherwise items with multiple classes would be wrongly categorized.

6.2.2 Selection of the Narrative Subject

When the user loads the narrative building tool for the first time, s/he is provided with a view that allows selecting the subject of the narrative. S/he can either select the subject from a few default examples, or insert the name of the person in a text field.

Figure 6.2 shows the Subject Selection view. The subject can be any entity that is present in Wikipedia. In the Figure we show a few default entities, including notable people from various historical periods, an organization (NASA), and a physical object (the Tower of Pisa). If the user selects one of these default entities, the narrative creation phase begins.

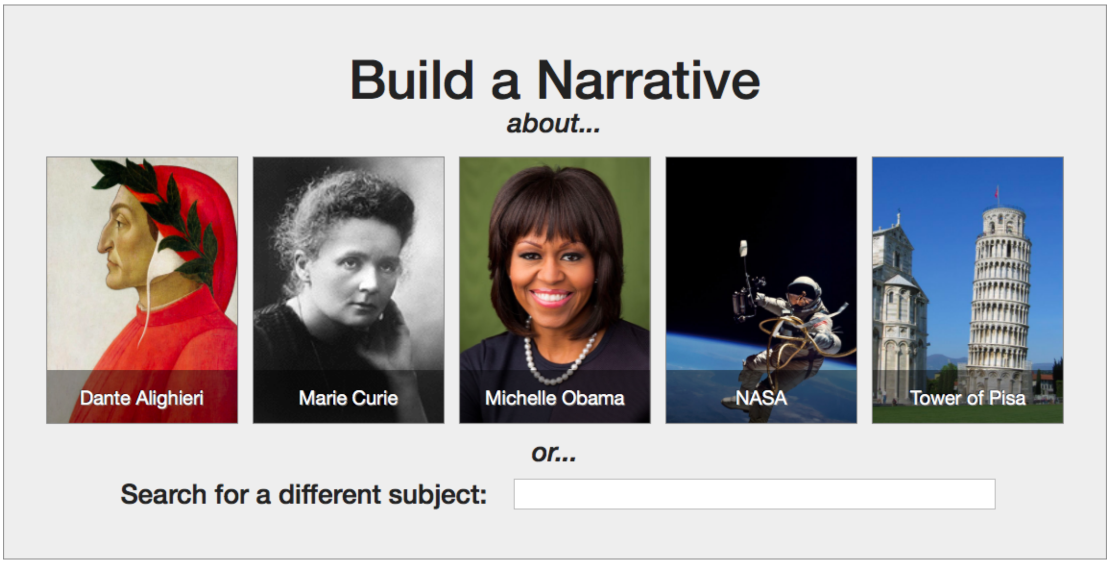


Figure 6.2: *View of the subject selection interface.*

If the user prefers to search for another subject, she can type some characters in order to be provided with a visual list of possible subjects whose names includes the inserted text, created by querying the Wikipedia API and visualized dynamically using the jQuery library. Clicking on the name of a person starts the narrative creation phase.

6.2.3 Creation of a Narrative

The second view of the tool allows creating a narrative. The interface is divided into three main parts:

1. The left-hand side of the screen contains a list of all entities that can be chosen as event components and that correspond to the properties that contextualize an event, as reported in Section 5.3.1 in Chapter 4. Furthermore it contains buttons to filter the displayed entities, search them by name, or add new ones.
2. The right-hand side of the screen contains the main event creation form, and a series of buttons to save the event, clear the form, or switch to the relations screen;
3. the bottom side of the screen contains the event timeline, which can be scrolled horizontally.

The narrative creation screen is shown in Figure 6.3.

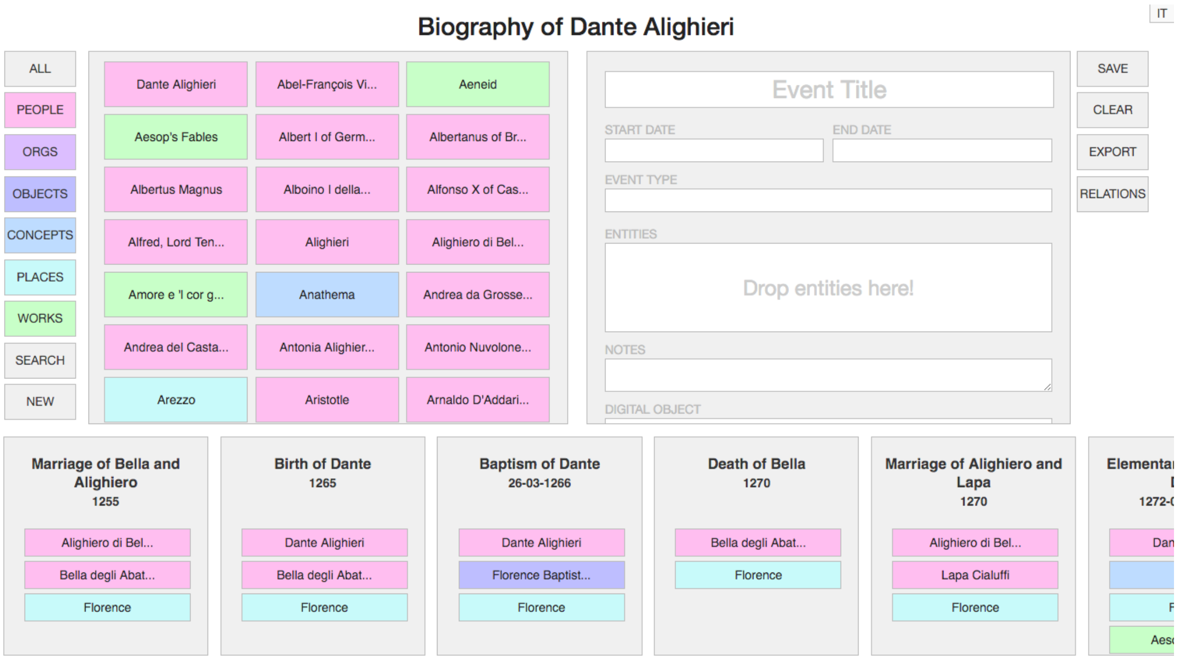


Figure 6.3: *View of the narrative creation interface.*

First Load and Entity List

During the first load, the tool performs several queries to the Wikipedia API and the Wikidata Query Service (WQS). First of all, it queries the Wikipedia API to extract all the Wikipedia pages that are linked to the one about the subject of the narrative. For instance, for Dante Alighieri it would extract pages about the members of his family, his works, the places where he lived, the people he interacted with, etc. Then, the tool queries the WQS for the names, descriptions, and classes of each of the Wikidata items corresponding to these Wikipedia pages, using the SPARQL query shown in Figure 6.4.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX schema: <http://schema.org/>
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
SELECT DISTINCT ?uri ?type ?name ?desc ?image
WHERE {
   ?uri wdt:P31 ?class.
   ?class wdt:P279* ?type.
   OPTIONAL { ?uri wdt:P18 ?image. }
   OPTIONAL { ?uri rdfs:label ?name FILTER (lang(?name) = "en"). }
   OPTIONAL { ?uri schema:description ?desc FILTER (lang(?desc) = "en"). }
   VALUES ?uri { [list of Wikidata entity IDs] }
   VALUES ?type { [list of Wikidata class IDs] }
}
```

Figure 6.4: *SPARQL query to extract entities from Wikidata.*

Once the previous step is concluded, the tool classifies the resulted entities using the Wikidata-CRM mapping discussed in Section 6.2.1. The six main classes and the corresponding mapping are reported in Table 6.1. At this point, the tool saves the list of

Wikidata	*Our ontology*
Q5 human	E21 Person
Q16334295 group of humans	E74 Group
Q7184903 abstract object	E89 Propositional Object
Q223557 physical object	E19 Physical Object
Q17334923 location	E53 Place
Q234460 text	E33 Linguistic Object
Q186081 time interval	E52 Time-Span
Q1190554 event	E5 Event

Table 6.1: *Mapping between the classes of Wikidata and those of the ontology of narratives.*

entities to an IndexedDB database instance[8] and displays them in the container on the left-hand side of the screen. Each entity is color-coded according to its class. The user can filter entities by clicking on the button corresponding to each class, or by clicking on the Search button. When clicking on the Search button, a search field appears on top of the list, allowing the user to filter the entities by name. A detailed view of the entities container is shown in Figure 6.5.

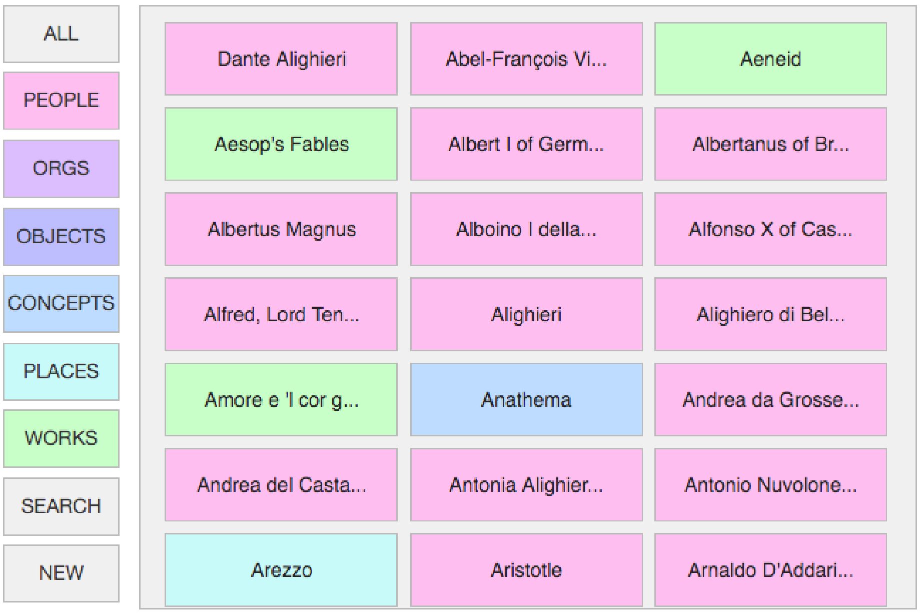


Figure 6.5: *View of the entities container.*

Clicking on each entity from the list, a *popover*[9] appears, showing the name, a description, an image of the entity loaded from the Wikimedia Commons[10] repository (when available), and links to the corresponding Wikipedia and Wikidata pages. An example of entity popover is shown in Figure 6.6. The user can also create new entities by clicking on the *New* button found on the left side of the entity container. A popover appears, where the user can insert:

- a Wikipedia URL, Wikidata URL (or URI), or Wikidata ID to automatically load

[8] https://developer.mozilla.org/it/docs/Web/API/IndexedDB_API
[9] The *popover* is an interface element provided by the Bootstrap library.
[10] http://commons.wikimedia.org

the entity from Wikidata, or,

- a name, description, and class to create a new entity.

The entity is then saved to the IndexedDB database and added to the main list.

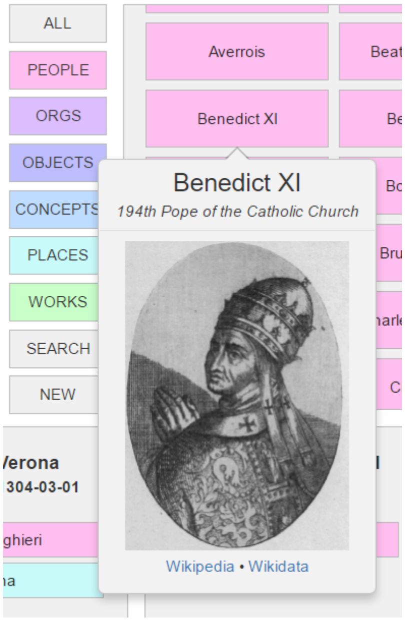


Figure 6.6: *An example of entity popover for Pope Benedict XI.*

6.2.4 Event Creation Form

The right-hand side of the screen contains the main event creation form. By filling in this form, the user can insert the basic data of each event. In particular, the uses can add:

- the event title (a string of text),
- the start date of the event (a string of text, automatically parsed),
- the end date of the event (a string of text, automatically parsed),
- the event type (a string of text, select from a list or added by the user),
- one or more Wikidata or user-defined entities that the user has identified as components of the event (added through drag-and-drop),
- an optional textual note (a string of text),
- an optional link to a digital object (a string of text, automatically parsed).

A detailed view of the event creation form is shown in Figure 6.7.

When the user begins creating an event, s/he writes the title of the event in the "Event Title" field of the event creation form. Then s/he types the start and end dates of the event, and she inserts an event type. The event types are loaded from the list, as explained in Section 6.2.5, but the user can also insert a new type. The following

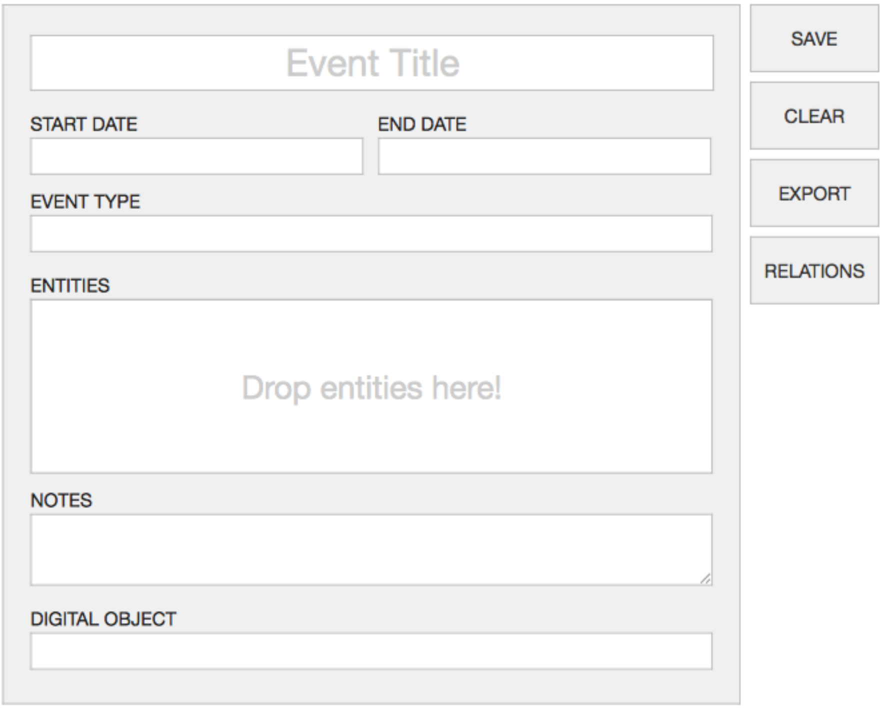


Figure 6.7: *View of the event creation form.*

step is the linking of the event to its components such as people, places, objects, etc. This activity can be done through a simple drag-and-drop, developed using the jQuery UI library. When the user drags an entity from the left-hand table onto the "Entities" field in the event creation form, a popover appears, showing the user more data entry options. This popover is shown in Figure 6.8.

This popover is used to enter sourcing information. The main sections in the popover are: Secondary source, Primary source(s), and Notes. In the Secondary source section, the user can insert the book, chapter, and fragment of the secondary source stating that this particular entity took part in the event. For instance, if the user is a biographer of Dante Alighieri, s/he will insert the book, chapter, and fragment of the biography s/he has written. In the Primary source section, the user can add one or more primary sources, including their author, title, the fragment of the secondary source that references it, the fragment of the primary source itself, its bibliographic references. To facilitate the addition of primary sources and to be sure that there are no duplicates, the primary source title and author fields have been constructed as auto-complete fields using the Typeahead.js library. When the user inserts the title of a work, s/he is provided with a list of all works in Wikidata with that title. If s/he selects a particular work, the author's name is filled automatically. If instead s/he starts by inserting a name in the author field, s/he is provided with a list of all authors found in Wikidata under that name. Upon selecting a particular author from the list, the title field automatically changes to show only the works by that author. In both the cases, the primary sources are stored internally along with their titles and corresponding Wikidata URIs (if available). Finally, the last section allows the user to insert a textual note about the sources or the relation of the entity to the event. If the entity is a person, the sourcing popover contains an

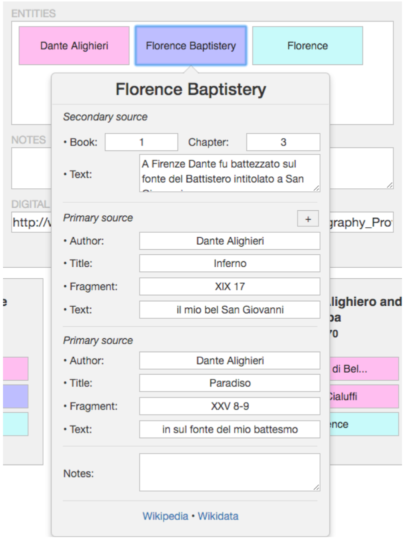


Figure 6.8: *View of the sourcing popover.*

additional section that allows the user to insert the role that the person played in the event (for instance, in an event of writing, the person who performed the writing will have role ""writer"). The role section is present only for people, since other classes of entity are not allowed to have a role. The role can be chosen from a list that is obtained through the process described in Section 6.2.7. The list is initially filtered based on the specific entity, e.g. a journalist who also wrote books would have "journalist" and "writer" as suggested roles. These role suggestions are obtained through SPARQL queries to the Wikidata knowledge base. An example is shown in Figure 6.9.

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
SELECT DISTINCT ?occupation
WHERE {
   OPTIONAL { wd:Q1067 wdt:P106 ?occupation. }
}
```

Figure 6.9: *Example of a SPARQL query to extract occupations for Q1067 (Dante Alighieri).*

Additional suggestions are based on the event type, e.g. if the event is of type "birth" the role menu will contain "mother", "father", and "child". However, the user can also bypass the suggestions and select a role from the full list described in 6.2.7 by selecting "Other" in the menu. When the role is undefined, it defaults to "participant". Once the user has filled the sourcing popover, s/he can close it and move on to another entity. After adding all related entities to the event, s/he can optionally add a textual note about the event and a link to a digital object. Then s/he can save the event or, in the case of mistake, can clear the form and start again using the buttons on the right-hand side of the event form.

6.2.5 Event Types

A list of types of event was identified in order to classify the events that compose the narratives. This is useful to (i) define recurring patterns in similar narratives, (ii) re-use these recurrent patterns in different narratives, only changing the instances of the involved classes, (iii) automatically extract the same types of events from different narratives on the same topic. In order to maximize the interoperability of the system, types of events defined in the CIDOC CRM standard ontology and in Wikidata were re-used. The following list reports the types of events extracted from the CRM and, when available, the equivalent Wikidata class:

- *E63 Beginning of Existence*, it may be used "for temporal reasoning about things (intellectual products, physical items, groups of people, living beings) beginning to exist; it serves as a hook to determine of a *terminus post quem* and *ante quem*"[11]. This is a subclass of *E5 Event*, and is equivalent to the Wikidata class *Q23956340 Beginning of Existence*. Its following subclasses were also used:
  - *E12 Production*: this class "comprises activities that are designed to, and succeed in, creating one or more new items", and is equivalent to the Wikidata class *Q739302 Production*.
  - *E65 Creation*: the class "comprises events that result in the creation of conceptual items or immaterial products, such as legends, poems, texts, music, images, movies, laws, types etc".
  - *E66 Formation*: this class "comprises events that result in the formation of a formal or informal E74 Group of people, such as a club, society, association, corporation or nation".
  - *E67 Birth*: this class "comprises the births of human beings. E67 Birth is a biological event focussing on the context of people coming into life", and is equivalent to the Wikidata class *Q34581 Birth*.
  - *E81 Transformation*: the class "comprises the events that result in the simultaneous destruction of one or more than one E77 Persistent Item and the creation of one or more than one E77 Persistent Item that preserves recognizable substance from the first one(s) but has fundamentally different nature and identity."
- *E64 End of Existence*, "it may be used for temporal reasoning about things (physical items, groups of people, living beings) ceasing to exist; it serves as a hook for determination of a terminus postquem and antequem". This is a subclass of *E5 Event*, and is equivalent to the Wikidata class *Q23956356 End of Existence*. The following subclasses were used:
  - *E6 Destruction*: this class "comprises events that destroy one or more instances of E18 Physical Thing such that they lose their identity as the subjects of documentation".
  - *E68 Dissolution*: this class "comprises the events that result in the formal or informal termination of an E74 Group of people".

[11] All quotations in this Section are taken from the CRM specifications: http://www.cidoc-crm.org/docs/cidoc_crm_version_6.2.1.pdf

  - *E69 Death*: this class "comprises the deaths of human beings", and corresponds to the Wikidata class *Q4 Death*.
  - *E81 Transformation*, defined as in the previous list of subclasses.

- *E85 Joining*, this class "comprises the activities that result in an instance of E39 Actor becoming a member of an instance of E74 Group. This class does not imply initiative by either party. It may be the initiative of a third party. Typical scenarios include becoming a member of a social organisation, becoming employee of a company, marriage, the adoption of a child by a family and the inauguration of somebody into an official position". This is a subclass of *E7 Activity*

- *E86 Leaving*, this class "comprises the activities that result in an instance of E39 Actor to be disassociated from an instance of E74 Group. This class does not imply initiative by either party. It may be the initiative of a third party. Typical scenarios include the termination of membership in a social organisation, ending the employment at a company, divorce, and the end of tenure of somebody in an official position". This is a subclass of *E7 Activity*

The following types of event, specific to biographies, cannot be found in the CRM, therefore they were imported from the Wikidata ontology:

- *Q35856 Baptism*, "Christian rite of admission and adoption, almost invariably with the use of water".
- *Q178561 Battle*, "part of a war which is well defined in duration, area and force commitment".
- *Q209715 Crowning*, "ceremony marking the formal investiture of a monarch and/or his or her consort with regal power".
- *Q476300 Competition*, "contest between organisms, animals, individuals, groups, etc."
- *Q93190 Divorce*, "termination of a marital union".
- *Q8434 Education*, "learning in which knowledge and skills is transferred through teaching".
- *Q40231 Election*, "process by which a population chooses an individual to hold public office"
- *Q188863 Exile*, "event by which a person is forced away from home"
- *Q1725430 Lawmaking*, "process of crafting legislation"
- *Q8445 Marriage*, "social union or legal contract between people called spouses that creates kinship".
- *Q2761147 Meeting*, "event in which two or more people assemble".
- *Q124734 Rebellion*, "refusal of obedience or order".
- *Q2359691 Residence*, "presence in a specific jurisdiction".

- *Q1763090 Sentence*, “decree of punishment in law”.
- *Q61509 Travel*, “movement of people between relatively distant geographical locations”.
- *Q198 War*, “organised and prolonged conflict between states”.

Types of events suitable for representing a biography were added. An enrichment of the types has been planned as future work, possibly taking their definitions from Wikidata, or from DBPedia or FreeBase. The final aim is to create a sufficiently large list of event types to extend the proposed system to other types of narratives.

6.2.6 Default Events

In the case of biographical narratives, several events can be inferred from the knowledge that is stored in Wikidata. For instance, a user describing the life of Dante Alighieri will probably have to insert an event titled “Death of Dante Alighieri” with a date of 14 September 1321. This information is already present in Wikidata, but since its ontology is not event-based, it is not represented as an event. Instead, Wikidata contains the statement “Q1067 (Dante Alighieri) P569 (date of death) 14 September 1321”, i.e. it uses the property *P569 date of death* to link the entity that represents Dante Alighieri to the date value “14 September 1321”. In order to facilitate the user’s work, a feature that extracts basic events from the knowledge contained in Wikidata and uses them to populate the timeline at the beginning of the narrative creation was developed. The extraction is performed through an initial SPARQL query to the Wikidata Query Service that retrieves RDF triples having the subject of the narrative as their subject. An example of this kind of SPARQL query is shown in 6.10.

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
SELECT DISTINCT ?birth ?death
WHERE {
   OPTIONAL { wd:Q1067 wdt:P569 ?birth. }
   OPTIONAL { wd:Q1067 wdt:P570 ?death. }
}
```

Figure 6.10: *Example of a SPARQL query to extract birth and death dates for Q1067 (Dante Alighieri).*

The query results are used to populate the timeline with several “default events”. The user is then able to easily edit the default events, or delete them if they are not needed in the narrative. Currently, the tool is able to handle basic life events such as birth, death, and marriage. The extension of the list of default events was planned as future work.

6.2.7 Roles

A list of roles to describe the way in which an actor participates in an event was defined . It should be noted that roles are sometimes dependent on the type of event, for instance an event of type *Q188863 Exile* will probably have an exiler and an exiled roles. These cases require custom definitions that are associated to each event type. Here is the full list of roles for the event types:

- Baptism: baptizer, baptized
- Battle: combatant
- Birth: mother, father, child
- Creation: creator
- Crowning: crowner, crowned
- Dissolution: dissolver
- Divorce: divorcee
- Education: student, teacher
- Election: elected, elector
- Friendship: friend
- Joining: member
- Lawmaking: lawmaker
- Marriage: spouse
- Production: producer
- Rebellion: rebel
- Residence: resident
- Sentence: sentencer, sentenced
- Transformation: transformer
- Travel: traveller

In other cases, roles can be expressed in a more general way. For instance, a meeting of the Chamber of Deputies will have exactly one participant with role "President of the Chamber of Deputies" and several participants with role "Deputy". This role is equivalent to the position *Q1055894 Deputy* from Wikidata. However, in Wikidata *Q4164871 Position* is a subclass of *Q12737077 Occupation*, which is a subclass of *Q1914636 Activity*. An activity is an event, not a role. For this reason, it was decided to not use Wikidata URIs directly for roles. Instead, the names of these occupations were imported but new URIs for them were defined, in order to use them as roles. To do so, the full list of instances of *Q12737077 Occupation* from Wikidata was extracted and it was used as a basis for the list of roles.

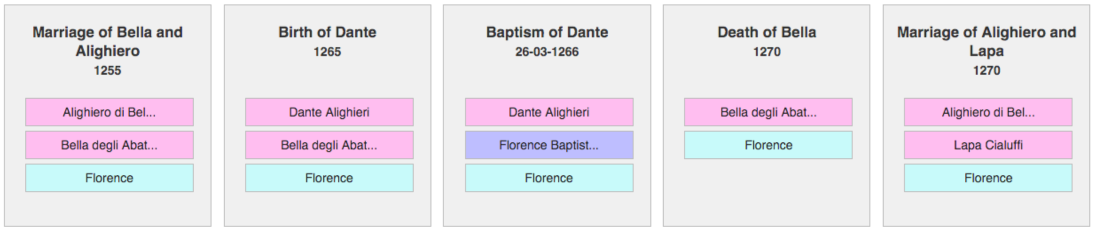


Figure 6.11: *View of the event timeline.*

6.2.8 Event Timeline

The bottom side of the screen is a timeline containing a simple view of each event created by the user in chronological order. The timeline is shown in Figure6.11. Whenever an event is saved or loaded as a default event, it is automatically added to the bottom timeline.

Each event in the timeline shows a reduced representation of the data entered by the user, i.e. the title, the dates, and the entities that are related to the event. By clicking on an event in the bottom timeline, the user can easily reload the full information about the event into the event creation form for subsequent editing or detailed visualization. The user can also delete events from the timeline by clicking on the Delete button in the top-right corner of each event.

6.2.9 Causal and Mereological Relations

The user can add relations among events by loading the third view of the interface, which is called "Causal and Mereological Relations" and can be accessed through a button on the right-hand side of the event creation form. In this view, shown in Figure 6.12, the user can link an event to the events that caused it, and also to those that contain it mereologically. These correspond to the relations defined in Section 4.2 of the Chapter 4.

When the user clicks on the Relations button, the bottom timeline slides up, revealing boxes where events can be freely dragged. These are taken from a duplicate timeline, which is put on the bottom of the screen. This way, the user can freely connect events to each other without losing sight of the bigger view. At any moment, the user can go back to the previous view using the Back button. All inserted relations are automatically saved in the IndexedDB database of the Web application.

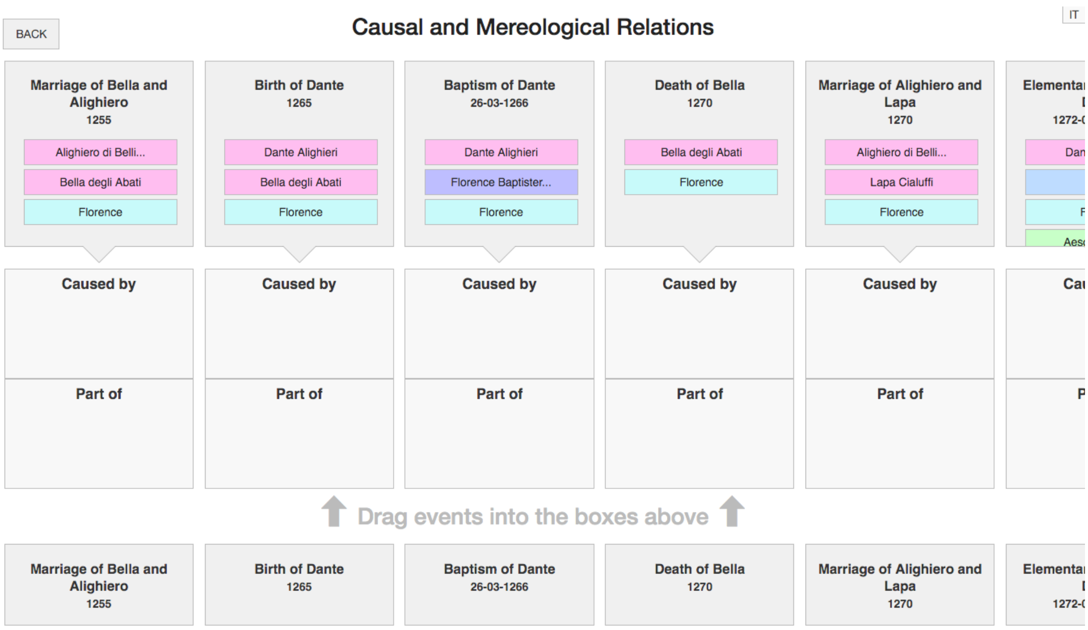


Figure 6.12: *Causal and mereological relations.*

6.2.10 Subsequent Loadings and Export

On subsequent loadings of the tool, all the data inserted by the user are automatically loaded from the IndexedDB database and the tool does not query neither Wikipedia nor Wikidata. The list of entities and the timeline of events are re-created locally and can be edited by the user anytime. Once the user has finished adding events and relations, s/he can download a JSON object containing the full list of events (including their relations) by using the "Export" button. The JSON object is an array containing all events in sequence. Each event has the following attributes:

- *id*: the numeric ID of the event
- *title*: the title of the event
- *type*: the type of the event
- *start*: the start date of the event
- *end*: the end date of the event
- *objurl*: optional URL of a digital object
- *notes*: optional textual notes
- *props*: a key/value map of propositions

Each proposition in the map is identified by a key corresponding to the Wikidata or user-defined ID of the related entity that constitutes the object of the proposition. Each proposition has the following attributes:

- *name*: the name of the related entity
- *class*: the class of the related entity

- *book*: optional book number of the secondary source
- *chapter*: optional chapter number of the secondary source
- *fragment*: optional fragment of the secondary source
- *notes*: optional textual notes
- *sources*: an array of primary sources

Each primary source has the following attributes:

- *authorId*: the Wikidata or user-defined ID of the author
- *authorName*: the name of the author
- *sourceId*: the Wikidata or user-defined ID of the primary source
- *sourceTitle*: the title of the primary source
- *textFrag*: optional textual fragment
- *refFrag*: optional reference fragment

The representation of an example event (the birth of Dante Alighieri) is shown in Figure 6.13.

6.2.11 Triplification of the Collected Knowledge

Once the user has completed the creation of the narrative, the events are organized in chronological order and are saved to a JSON object along with the entities that compose them. A Java software was developed to automatically translate the JSON object to an OWL (Web Ontology Language) graph, in which the knowledge is represented as triples subject-predicate-object. First of all, the software (SW) imports a JSON file and converts each event, related entity and relation between events to a Java object. These Java objects are organized into three lists: (i) a list of events, (ii) a list of related entities, and (iii) a list of relations between events. The SW uses the Apache Jena framework[12] that provides a programmatic environment for OWL. Jena is an open source Semantic Web framework for Java. It provides an API to extract data from and write to RDF and OWL graphs. The graphs are represented as an abstract model. A model can be sourced with data from files, databases, URIs or a combination of these. A Jena model was defined (see Appendix A), i.e. a grammar that describes the structure of the ontology. The classes and properties that compose the model, including those that were not present in the reference ontologies or were added as news ones, were defined. For each class, an OWL resource was defined, and for each property the domain and range were indicated. On the three lists the model was applied, which allows creating an OWL representation of the Java objects. The resulting graph was saved to a file in RDF/XML[13] and Turtle[14] formats. Then, the OWL graph was imported into a Virtuoso triple store [39] using the Virtuoso Jena Provider[15]. Since the CIDOC CRM was used as reference ontology, the full OWL representation of the CRM was imported into Virtuoso. Erlangen

[12] `https://jena.apache.org/`
[13] `https://www.w3.org/TR/rdf-syntax-grammar/`
[14] `https://www.w3.org/TR/turtle/`
[15] http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtJenaProvider

```
{
   "id": "1",
   "title": "Birth of Dante Alighieri"
   "type": "birth",
   "start": "1265",
   "end": "1265",
   "book": "1",
   "chapter": "3",
   "fragment": "Certainly Dante was born in Florence",
   "notes": null,
   "objurl": null,
   "props": {
      "Q1067": {
         "name": "Dante Alighieri",
         "class": "person",
         "role": "child",
         "notes": "",
         "sources": []
      },
      "Q3637724": {
         "name": "Bella degli Abati",
         "class": "person",
         "role": "mother",
         "notes": "",
         "sources": []
      }
      "Q2044": {
         "name": "Florence",
         "class": "place",
         "sources": [
            {
               "authorId": "Q1067",
               "authorName": "Dante Alighieri",
               "sourceId": "Q4509219",
               "sourceTitle": "Inferno",
               "textFrag": "io fui nato e cresciuto sovra ’l bel
                              fiume d’Arno, alla gran villa",
               "refFrag": "XXIII 94-95"
            }
         ]
      }
   }
}
```


Figure 6.13: *JSON representation of the event “Birth of Dante”.*

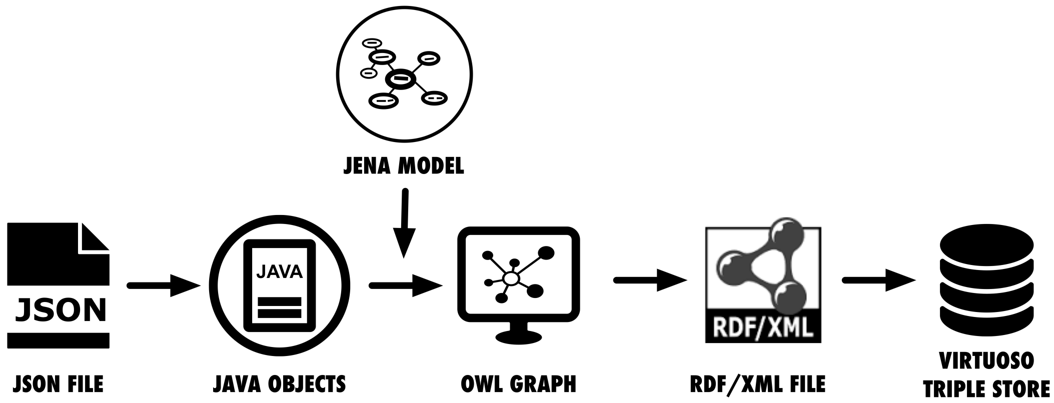


Figure 6.14: *The representation of the entire process to create the OWL graph.*

CRM/OWL[16], an OWL-DL 1.0 implementation of the CRM [53] was adopted. Erlangen CRM / OWL is an interpretation of the CRM in a logical framework attempting to be as close as possible to the text of the specification. The Erlangen implementation was developed by Bernhard Schiemann, Martin Oischinger and Günther Görz at the Friedrich-Alexander-University of Erlangen-Nuremberg, Department of Computer Science, Chair of Computer Science 8 (Artificial Intelligence) in cooperation with the Department of Museum Informatics at the Germanisches Nationalmuseum Nuremberg and the Department of Biodiversity Informatics at the Zoologisches Forschungsmuseum Alexander Koenig Bonn. Figure 6.14 schematically shows the entire process to create the OWL graph.

[16]http://erlangen-crm.org/

CHAPTER 7

Evaluation

An initial qualitative evaluation of the developed ontology and tool is presented in this Chapter. The evaluation is based on the semantic network representing the narrative of Dante's life, described in detail in Chapter OntologyValidation (Section 5.3.1). In particular, the evaluation concerned the following aspects:

- the representational adequacy of the ontology by a historian who is an expert of Dante Alighieri's life;
- the effectiveness of a tool to build narratives;
- the satisfaction of the users' requirements, defined at the beginning of the study (see Chapter 3)

Most of the effort was concentrated on the last point, while the first two evaluations are preliminary.

7.1 Representational Adequacy of The Ontology

The experiment to build a formal narration of Dante's life allowed performing a first evaluation of the representational adequacy of the ontology. During the creation of the narrative, the problem of finding the appropriate classes and properties of the ontology was faced, in order to formally represent the content of the biography. This formal representation was performed without changing the ontology. Since the ontology directly reflects the structure of a narrative as defined in narratology, this can be considered as an indication that narratology has reached a certain maturity, and that the structure of narratives as defined by narratology is solid, in spite of the fact that certain notions remain difficult to formalize.

A second and more thorough evaluation of the semantic network was performed by the expert who provided the biography of Dante used to build the network. The exam was organized in a dialogic fashion: a brief list of questions was given to the expert, each addressing a specific aspect of this representation.

1. As first question, the expert was asked to evaluate the representational adequacy of the ontology, which is the ability of the ontology to capture, in a formal way, the salient aspects of the provided narrative as well as its provenance. To this end, the expert explored the narrative on the timeline and looked at the contextualization of the individual events of the fabula. The evaluation was *positive* and the scholar confirmed that the ontology was able to express all the formalizable knowledge related to the events and their relations, as described in the analysed text.

2. The second question was about the usefulness of using external resources to enrich the narrative. The scholar appreciated the possibility to extend the knowledge on each event using related resources, such as Wikipedia pages, digital objects descriptions relating to the event and included in external digital libraries such as Europeana, and related images extracted from Wikimedia Commons[1]. In his opinion, the semantic network shape of the knowledge could be exploited to make the content of the network easy to consume also for the non-experts.

3. As third question, the expert was asked to evaluate the representation of the provenance of the knowledge. In his opinion, the provenance gives crucial technical information to a scholar, e.g. the primary sources supporting an event. Having this information available allows supporting the narrative with evidence and allows readers to verify the trustworthiness of the biographical reconstruction.

4. Finally, as last question the expert was asked to evaluate the possibility to have different narratives of the same topic, created by different scholars using the tool. He answered that this allows comparing results thereby identifying common points and differences in the created narratives. In particular, the investigation of the different primary sources used to identify the events of the fabula could be very interesting in a historical reconstruction point of view.

7.2 Effectiveness of the narrative building tool

Measuring the effectiveness of the narrative building tool means to estimate how well the tool allows a user to build a formal narrative in a short time and with little effort.

As already mentioned, the narrative of Dante's life composed of 80 events was built in about five hours of work. It could be considered a very positive outcome since the work to construct the narrative had nothing to start with, apart from the narration in natural language and the knowledge base embodied in the tool. The expert considered as very encouraging the fact that it is possible to formalize a significant narrative from the text in a matter of a few hours. This performance was possible since the tool is provided with a semi-automatic knowledge extraction method that identifies candidate relevant named entities within the narration and automatically imports them from Wikidata. Indeed, 60% of the entities used in the narration of the life of Dante are those defined in Wikidata.

[1] https://commons.wikimedia.org

As future work, the improvement of the knowledge extraction method has been planned, adding more knowledge bases for extracting relevant entities. Natural candidates to this aim are DBPedia[2] and FreeBase[3]. Due to the positive results of this first evaluation experiment, it has been planned to make the tool available to a community of scholars in the context of an Italian national research project[4], in order to perform a larger scale evaluation.

7.3 Satisfaction of users' requirements

In order to proof the satisfaction of each requirement defined at the beginning of this research (see Chapter 3), a corresponding SPARQL query was implemented returning exactly what the requirement asked for. In this way, the ontology is potentially able to satisfy all the requirements. In particular, the requirements to consider at this stage of the project are: (1) those relating to the knowledge to extract from the knowledge base, and (2) those relating to the knowledge to be browsed via a Web interface. For convenience, the list of the main requirements is reported here below:

Knowledge to be extracted from the knowledge base:

- Events along with their primary sources
- Events involving certain given entities (e.g. place, person)
- Events happened in a certain given range of time
- Events linked by certain given relations (e.g. causal or mereological)

Developing a Web-based browsing functionality that allows:

- visualizing the fabula of a narrative on a timeline
- visualizing events (all or only someone defined by the user) along with their primary sources in format of table, exportable in CSV format
- visualizing events happened in a specific range of time (defined by the user) in format of table, exportable in CSV format
- using network graphs to visualize an event and its related entities
- using network graphs to visualize a certain given entity and its related events
- using network graphs to visualize the different types of relations that connect events

In the following, a complete description of the requirements and of the corresponding queries is reported. For each query, a specific visualization is defined in order to display the data in an appropriate way. In rest of the chapter, these visualization components are described.

[2] http://wiki.dbpedia.org/
[3] https://datahub.io/it/dataset/freebase
[4] http://perunaenciclopediadantescadigitale.eu

7.3.1 Timeline Visualization

First of all, in order to give a complete overview of the narrative, the events that compose its fabula were placed on a timeline, as shown in Figure 7.1. TimelineJS[5] was used for the implementation. This Javascript library allows visually navigating the semantic network of events defined for Dante's life.

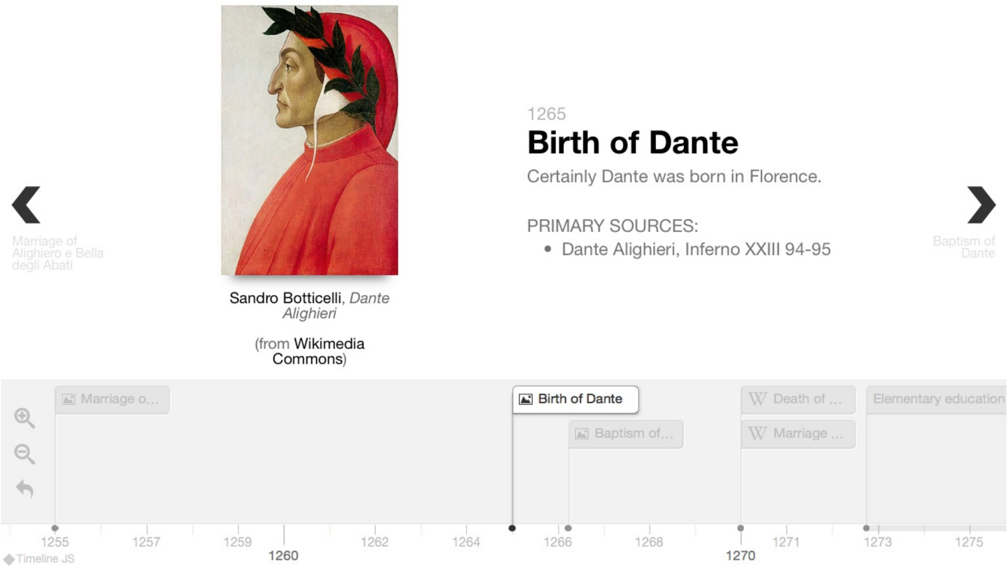


Figure 7.1: *Visualization of the events on a timeline.*

In particular, requirements for the tool are to extract the following information for each event on the timeline:

1. the title of the event ;
2. the date of the event;
3. the fragment of the secondary source, if present;
4. the digital object representing the event, if present;
5. the list of the primary sources, if present ;
6. an image describing one of the related entities, if available, or otherwise a widget linking to the Wikipedia page which describes the entity.

The information from point 1 to 5 can be directly extracted from the knowledge base using a SPARQL query. This query is shown in Figure 7.2. As reported in the SELECT clause, the query extracts the knowledge in this order: (i) the title of the event, (ii) the date in which the event starts and (iii) ends, (iv) the textual fragment of the written biography that describes the event (it is called the secondary source), (v) the title of the primary source, (vi) the textual fragment of the primary source, (vii) the reference of the primary source as reported in the secondary source, e.g. Inferno V, 1-9, (vii) the URI of the digital object that represents the event.

[5] https://timeline.knightlab.com

```
      PREFIX cnt:   <http://www.w3.org/2011/content#>
PREFIX nar:   <http://dantesources.org/narratives/narrativesRDFS.rdf#>
PREFIX crm:   <http://erlangen-crm.org/current/>

SELECT ?eventTitle ?startDate ?endDate ?fragmentText
      ?primarySourceTitle ?primaryTextFrag ?reference ?URIdigitalObject
FROM <http://dantesources.org/narratives>
WHERE {
   ?eventURI crm:P149_is_identified_by ?eventAppellation.
               ?eventAppellation crm:P3_has_note ?eventTitle.
               ?URIdigitalObject crm:P138_represents ?eventURI.
   ?eventURI crm:P4_has_time-span ?timeSpan.
   ?timeSpan crm:P79_beginning_is_qualified_by ?startDate.
   ?timeSpan crm:P80_end_is_qualified_by ?endDate.
   ?propositionURI rdf:subject ?eventURI.
   ?propositionURI rdf:object ?entityURI.
   ?exprFrag crm:P129_is_about ?propositionURI.
   ?exprFrag cnt:chars ?fragmentText.
   ?propositionURI nar:hasSource ?primarySourceURI.
   ?creationEvent crm:P94_has_created ?primarySourceURI.
   ?primarySourceURI crm:P149_is_identified_by ?primarySourceAppellation.
   ?primarySourceAppellation crm:P3_has_note ?primarySourceTitle.
   ?propositionURI nar:hasTextFragment ?primaryTextFrag.
   ?primarySourceURI crm:P106_is_composed_of ?primaryTextFrag.
   ?primaryTextFrag nar:hasReference ?reference.}
```

Figure 7.2: *SPARQL query to extract all knowledge shown in the timeline.*

Figure 7.3 shows an example of the visualization of a digital object, extracted through the query reported in Figure 7.2. On the right of the Figure 7.3 is shown the textual content of the secondary source. Below this text, a link to a digital object contained in an external digital library is reported. Clicking on the link it is possible to visualize a preview of the digital object (in this case, the Dante's Epistle XII to Cangrande della Scala).

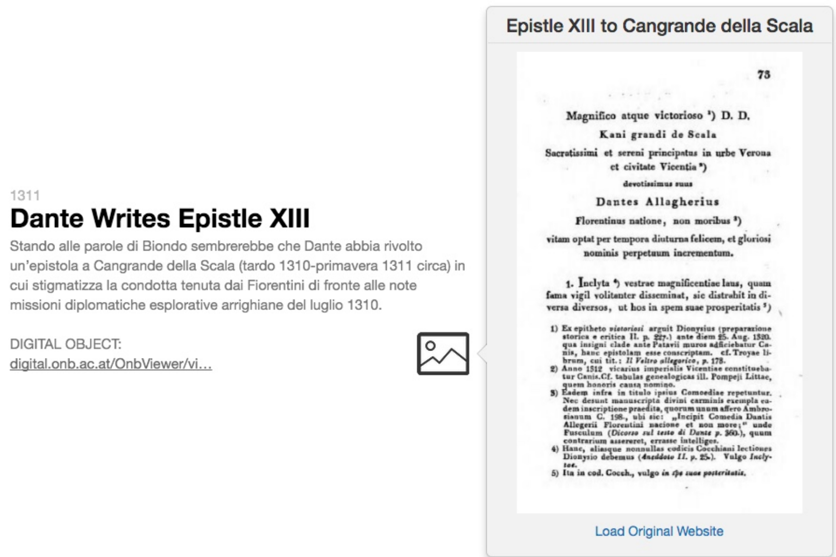


Figure 7.3: *Visualization of a digital object in the timeline.*

As far as the last requirements is concerned, i.e. the image or the reference of a Wikipedia page describing the entity, this information can be extracted from the knowledge base using a number of steps that were implemented and included as output of the visualised narrative:

1. A SPARQL query on the Wikidata Query Service to extract all available images of Wikidata entities that are present in the narrative. These images are presented in the Wikimedia Commons[6] repository. If the image is present, it is loaded from Wikimedia Commons. This SPARQL query is shown in Figure 7.4.

2. A SPARQL query on the Wikidata Query Service to extract links to Wikipedia pages describing the entities. This SPARQL query is shown in Figure 7.5.

3. For each event, the visualisation output shows the first image retrieved from the results of the query. An example is shown in Figure 7.6. On the left of the Figure, the textual content that describe the event along with a link to a corresponding digital object and two references to the primary sources are reported. On the right, an image that represents the event is shown, extracted from Wikimedia Commons. If no image is available, the process loads the corresponding Wikipedia page as shown in Figure 7.7.

```
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
SELECT DISTINCT ?entity ?image
WHERE {
   ?entity wdt:P18 ?image.
}
VALUES {?entity [list of Wikidata entities]}
```

Figure 7.4: *SPARQL query to extract images from Wikidata.*

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX schema: <http://schema.org/>
SELECT ?url
WHERE
{
   ?url schema:about wd:Q1067.
   ?url schema:inLanguage "en".
   ?url schema:isPartOf <https://en.wikipedia.org/>.
}
VALUES {?uri [list of Wikidata entities]}
```

Figure 7.5: *SPARQL query to extract links to the English Wikipedia pages related to the entities.*

[6] https://commons.wikimedia.org

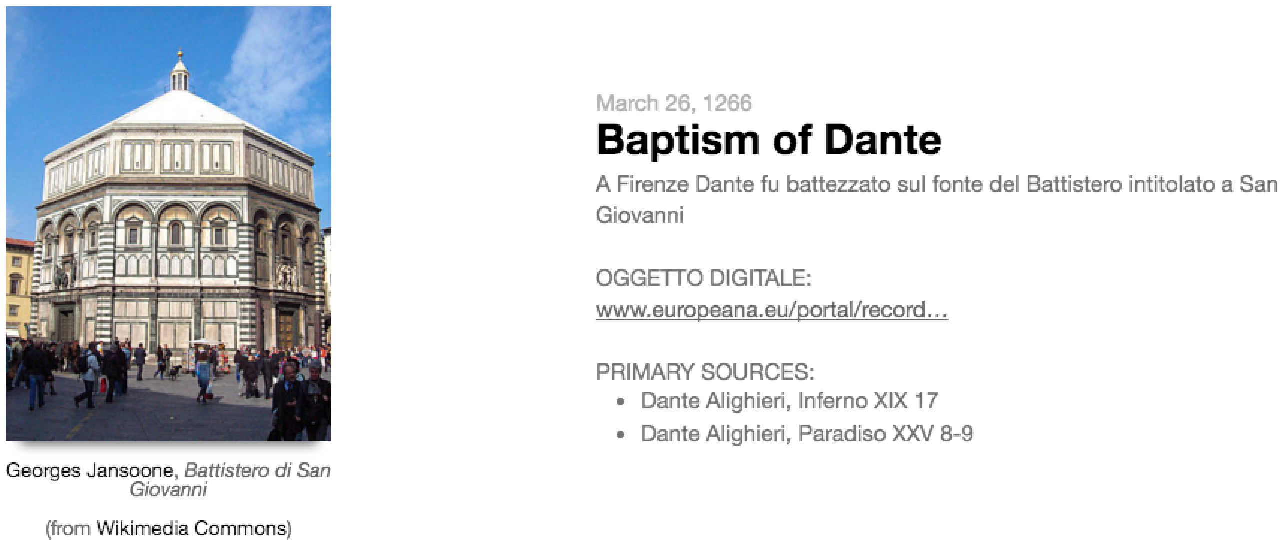


Figure 7.6: *Visualization of an image in the timeline.*

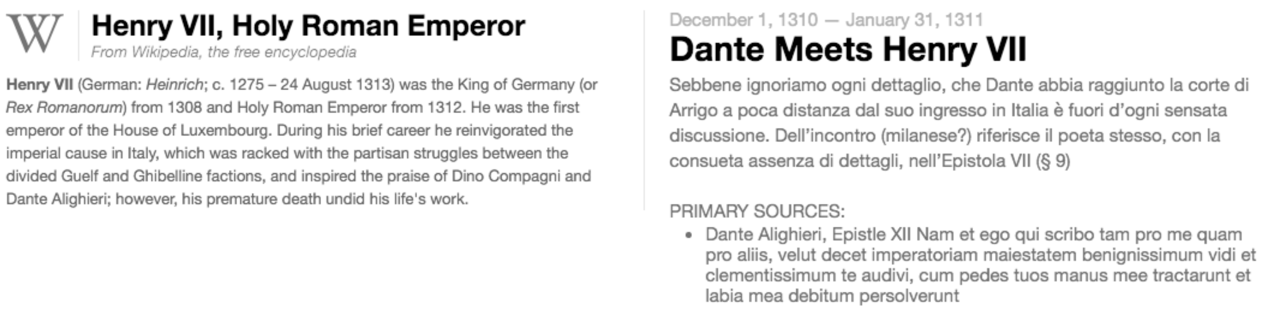


Figure 7.7: *Visualization of a Wikipedia page in the timeline.*

7.3.2 Event-Centered Graph

Another requirement for the tool is the visualization of the entities that compose each event. To this aim, a SPARQL query to extract this information from the knowledge base was implemented. This query is shown in Figure 7.8. As indicated in the SELECT clause, this query extracts, for each event title, the names and URIs of the corresponding entities.

```
PREFIX rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX crm:   <http://erlangen-crm.org/current/>

SELECT ?eventTitle ?entityURI ?entityName
FROM <http://dantesources.org/narratives>
WHERE {
   ?eventURI crm:P149_is_identified_by ?eventAppellation.
   ?eventAppellation crm:P3_has_note ?eventTitle.
   ?propositionURI rdf:subject ?eventURI.
   ?propositionURI rdf:object ?entityURI.
   ?entityURI crm:P149_is_identified_by ?entityAppellation.
   ?entityAppellation crm:P3_has_note ?entityName.
}
```

Figure 7.8: *SPARQL query to extract all entities of all events that compose the narrative.*

One example of the visualization of an event along with its related entities is shown in Figure 7.9. To this aim, the vis.js[7] Javascript library to implement the visualization was used. The network graph visualizes a star with the event in its centre; the related entities are represented with different icons depending on their ontological classes. The entities are connected to the event at the same level. By clicking on an entity, the user can visualize the corresponding Wikipedia page or, for the user-created entities, the description of the entity stored in the knowledge base.

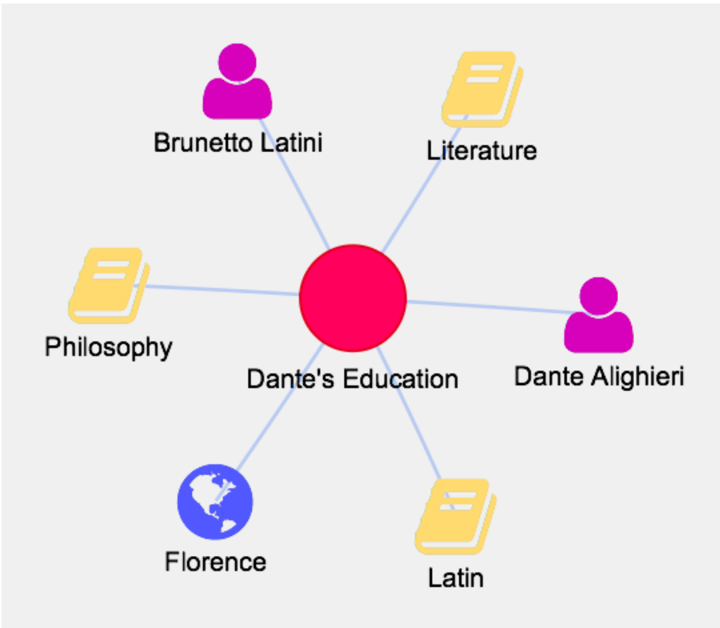


Figure 7.9: *Visualization of the event "Education of Dante" and its related entities.*

7.3.3 Entity-Centered Graph

In order to extract all the events that are related to a specific entity, a SPARQL query was developed and the result is presented as a vis.js network graph. The query is shown in Figure 7.10. This query extracts (see SELECT clause) all the events (URIs and titles)

[7] http://visjs.org

that occurred in the city of Florence. Indeed, the string "Florence" is set as value of the range of the property *P3 has_note* in the query.

```
PREFIX rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX crm:   <http://erlangen-crm.org/current/>

SELECT ?eventURI ?eventTitle
FROM <http://dantesources.org/narratives>
WHERE {
   ?eventURI crm:P149_is_identified_by ?eventAppellation.
   ?eventAppellation crm:P3_has_note ?eventTitle.
   ?propositionURI rdf:subject ?eventURI.
   ?propositionURI rdf:object ?entityURI.
   ?entityURI crm:P149_is_identified_by ?entityAppellation.
   ?entityAppellation crm:P3_has_note "Florence".
}
```

Figure 7.10: *SPARQL query to extract all entities of all events that compose the narrative.*

One example of visualization of an entity with its related events is reported in Figure 7.11. In the centre of the graph the entity "Florence" is shown, which is an instance of class *crm:E53 Place*. The edges connect this entity to all the events that took place in Florence. The Florence entity is represented with a circle along with its title.

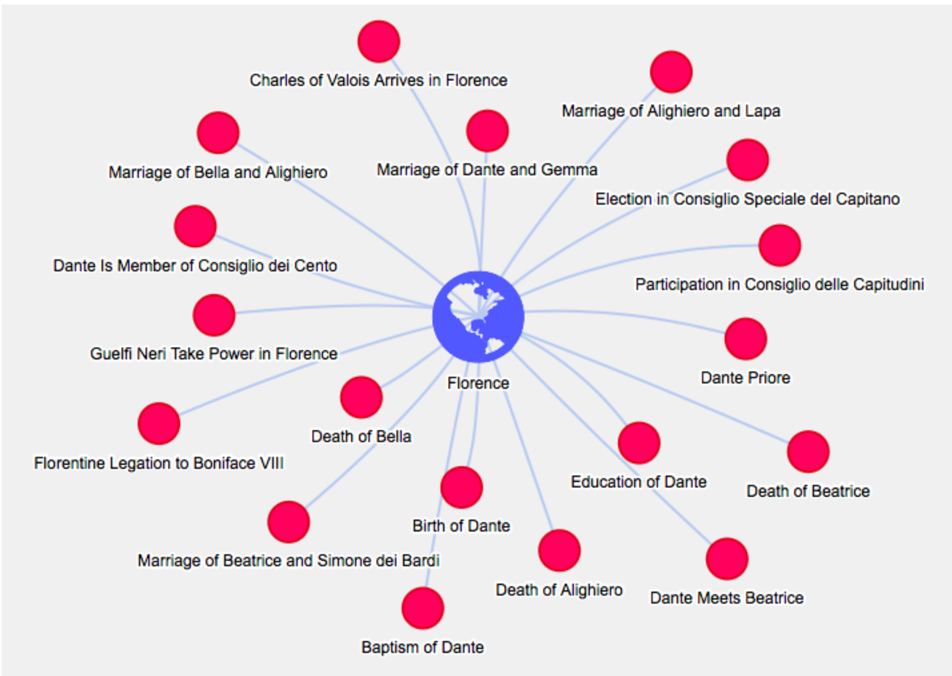


Figure 7.11: *Visualization of the entity "Florence" and its related events.*

7.3.4 Primary Sources Table

One of the most important requirements for a scholar who studies historical events, is the knowledge of their primary sources. In the case study, for each event, the user can add the following information about primary sources:

- the title of the primary source;

- the author of the primary source;
- the textual fragment of the primary source that describes the event;
- the reference of the textual fragment.

These pieces of knowledge can be extracted from the knowledge base using the SPARQL query shown in Figure 7.12. As reported in the SELECT clause, the query extracts: (i) the title of event, (ii) the date of the event, (ii) the title of the corresponding primary source, (iii) the name of the author of the primary source, (iv) the textual fragment of the primary source, (v) the reference of the primary source.

```
PREFIX rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX nar:   <http://dantesources.org/narratives/narrativesRDFS.rdf#>
PREFIX crm:   <http://erlangen-crm.org/current/>
PREFIX cnt:   <http://www.w3.org/2011/content#>

SELECT DISTINCT ?eventTitle ?startDate ?endDate ?primarySourceTitle
?authorName ?text ?reference
FROM <http://dantesources.org/narratives>
WHERE {
?eventURI crm:P149_is_identified_by ?eventAppellation.
?eventAppellation crm:P3_has_note ?eventTitle.
?propositionURI rdf:subject ?eventURI.
?propositionURI nar:hasSource ?primarySourceURI.
OPTIONAL {
   ?propositionURI nar:hasTextFragment ?textFragment.
         ?textFragment cnt:chars ?text.
         ?primarySourceURI crm:P106_is_composed_of ?textFragment.}
OPTIONAL {
   ?textFragment nar:hasReference ?refFragment.
   ?refFragment cnt:chars ?reference.
                 }
?creationEvent crm:P94_has_created ?primarySourceURI.
?creationEvent crm:P14_carried_out_by ?authorURI.
?primarySourceURI crm:P149_is_identified_by ?primarySourceAppellation.
?primarySourceAppellation crm:P3_has_note ?primarySourceTitle.
?authorURI crm:P149_is_identified_by ?authorAppellation.
?authorAppellation crm:P3_has_note ?authorName.
?eventURI crm:P4_has_time-span ?timeSpan.
?timeSpan crm:P79_beginning_is_qualified_by ?startDate.
?timeSpan crm:P80_end_is_qualified_by ?endDate.
              }
```

Figure 7.12: *SPARQL query to extract all events with their dates and their primary sources.*

The result of the query is visualized in table format in Figure 7.13. When an event has more than one primary source, the table shows a row for each source, e.g. the event "Baptism of Dante" has two primary sources: Dante's Inferno XIX 17 and Paradiso XXV 8-9. It is possible that the reference fragment is absent, for example when the primary source has no internal subsections, e.g. Boccaccio's work "Trattatello in laude di Dante".

Event	Date	Primary Source	Author	Text Fragment	Reference Fragment
Birth of Dante	1265	Inferno	Dante Alighieri	"io fui nato e cresciuto sovra 'l bel fiume d'Arno, alla gran villa"	XXIII 94-95
Baptism of Dante	1266-03-26	Inferno	Dante Alighieri	"il mio bel San Giovanni"	XIX 17
Baptism of Dante	1266-03-26	Paradiso	Dante Alighieri	"in sul fonte del mio battesmo"	XXV 8-9
Marriage of Dante and Gemma	1277-02-09	Instrumentum Dotis	Uguccione di Baldovino	"Domine Gemme vidue, uxori olim Dantis Allagherii et filie condam domini Manetti domini Donati, pro iure sue dotis librarum CC florenorum parvorum ut de instrumento dotis constat manu ser Ranaldi filii condam Oberti Baldovini de Florentia notarii, ex inbreviaturis ser Uguiccionis Baldovini notarii, facto in anno Domini M° CCLXXVI, indictione VJ, die VIIIJ° mensis februarii"	Capitani di Parte Guelfa, Numeri Rossi, 42, c. 3r
Dante's First Travel to Bologna	1286–1287	Trattatello in laude di Dante	Giovanni Boccaccio	"Egli li primi inizii, sì come di sopra è dichiarato, prese nella propia patria, e di quella, sì come a luogo più fertile di tal cibo, n'andò a Bologna"	

Figure 7.13: *Visualization of the table of the events along with their primary sources.*

7.3.5 Table of Events That Occurred in a Specific Time Range

The user has the possibility to select a specific range of time to visualize all events that occurred in that period. The user can freely insert the dates using a widget to select a full date or the year only. In order to be sure that the query always returns a result, the initial and end dates of the narrative are suggested. Figure 7.14 shows the SPARQL query to extract the events in Dante's life that occurred from 1265 to 1277. This range of time is set using the FILTER clauses in the query.

```
PREFIX crm:   <http://erlangen-crm.org/current/>

SELECT ?eventTitle ?startDate ?endDate
FROM <http://dantesources.org/narratives>
WHERE {
   ?eventURI crm:P149_is_identified_by ?eventAppellation.
   ?eventAppellation crm:P3_has_note ?eventTitle.
   ?eventURI crm:P4_has_time-span ?timeSpan.
   ?timeSpan crm:P79_beginning_is_qualified_by ?startDate.
   ?timeSpan crm:P80_end_is_qualified_by ?endDate.

   FILTER (xsd:integer(substr(?startDate, 1, 4)) > 1265).
   FILTER (xsd:integer(substr(?endDate, 1, 4)) < 1277).
}
ORDER BY ?startDate
```

Figure 7.14: *SPARQL query to extract all events that occurred from 1265 to 1277.*

The results of the query are shown in form of table, where for each event its dates are shown.

Event	Date
Baptism of Dante	1266-03-26
Death of Bella	1270
Marriage of Alighiero and Lapa	1270
Elementary Education of Dante	1272–1277
Death of Alighiero	1276
Marriage of Dante and Gemma	1277-02-09

Figure 7.15: *Visualization of the events that occurred in a selected time range.*

7.3.6 Relations Visualization

As reported in Chapter 4 where the ontology was described, three types of relations between events were defined: mereological, temporal and causal. A visualization component that shows these relations using a network graph was developed. The query in Figure 7.16 performs the extraction of all sub-events that compose the super-event "Education of Dante", which is set as value of the range of the property *P3 has_note* in the query. Indeed, the education of Dante is composed by three different events: elementary school, middle school and education received under his preceptor Brunetto Latini.

```
PREFIX crm:   <http://erlangen-crm.org/current/>

SELECT ?event1Title ?event2Title
FROM <http://dantesources.org/narratives>
WHERE {
        ?event1 ?crm:P9_consists_of ?event2
        ?event1 crm:P149_is_identified_by ?eventAppellation.
        ?eventAppellation crm:P3_has_note "Education of Dante".
        ?event2 crm:P149_is_identified_by ?eventAppellation.
        ?eventAppellation crm:P3_has_note ?event2Title.
}
```

Figure 7.16: *SPARQL query to extract all sub-events that compose the super-event ""Education of Dante".*

The mereological relations between the event "Education of Dante" and its sub-events are shown in Figure 7.17. The figure also shows the temporal relations that occur between the sub-events. Also in this case, vis.js was used to implement the graph.

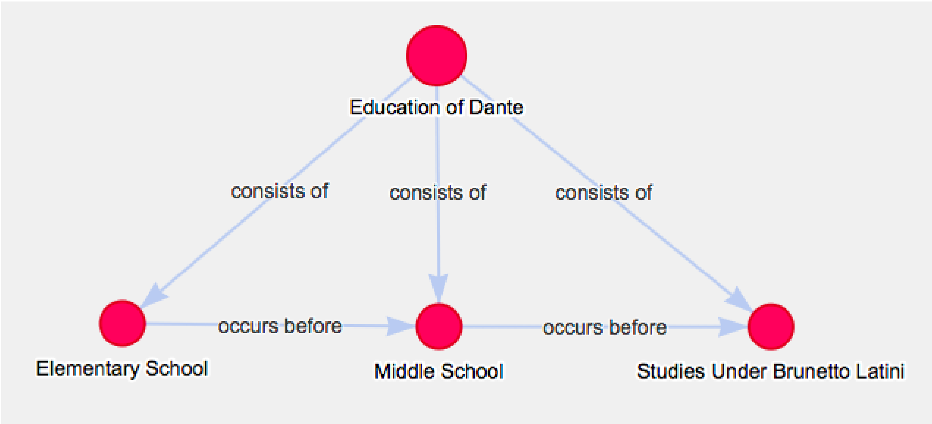


Figure 7.17: *Visualization of the events that occurred in a selected time range.*

CHAPTER 8

Conclusions

8.1 Summary

One of the main problems of the current Digital Libraries (DLs) is the limitation of the informative services offered to the user who aims at discovering the resources of the DL by asking requests (queries) in natural language. Indeed, all DLs provide simple search functionalities that return a ranked list of their resources. No semantic relation among the returned objects is usually reported that could help the user to obtain a more complete knowledge on the subject of the search. The introduction of the Semantic Web, and in particular of the Linked Data, has the potential of improving the search functionalities of DLs. In this context, the long-term aim of this thesis has been to introduce the narrative as new first-class search functionality. As output of a query, the envisaged new search functionality should not only return a list of objects but it should also present one or more narratives, composed of events that are linked to the objects of the existing libraries (e.g. Europeana) and are endowed with a set of semantic relations connecting these events into a meaningful semantic network. As a necessary step towards this direction, the thesis has presented an ontology of narratives along with a tool for the construction of narratives based on the ontology. Moreover, it has used to the tool for evaluating the ontology in the context of an experiment centered on the biography of the Italian poet Dante Alighieri. More specifically:

- An overview of the related works developed in the Semantic Web field and in Narratology, and especially in its sub branch named Computational Narratology was reported. In particular, the basic principles of Narratology and their application of the Artificial Intelligence field have been reviewed along with the study of the Artificial Intelligence literature, especially of the Event Calculus theory, in order to identify the formal components of narratives. Further, the adopted methodological approach and the research requirements have been explained and reported in detail.

- A conceptualization of narratives has been developed, based on notions derived from narratology and Artificial Intelligence. According to this conceptualization, a narrative consists of a fabula and several narrations of this fabula (plots), linked to the fabula by an event association relation. A mathematical expression of the conceptualization has been given, in order to provide a characterization of the conceptualization as clear and as precise as possible, also to be used as a basis for the subsequent development of an ontology of narratives, encoded in OWL. The proposed conceptualization has been validated by expressing it into an existing ontology, the CIDOC CRM, and by endowing it with provenance knowledge, also expressed in a derivation of the CRM named CRMinf. This expression has been used in the validation experiment, consisting in the modelling a narrative of the life of Dante Alighieri, provided by a biographer who has scientifically supported this research.

- The population of the created ontology has been performed by means of a semi-automatic approach implemented by a tool for the construction of narratives which obey the ontology. This tool retrieves and assigns URIs to the instances of the classes of the ontology using an external knowledge base as resource (Wikidata) and also facilitates the construction and contextualization of events, and their linking to form the fabulae of narratives.

- A qualitative validation of the developed ontology has been finally carried out. This validation has regarded the evaluation of: (i) the representational adequacy of the ontology by a Dante Alighieri's expert; (ii) the effectiveness of the narrative building tool; (iii) the satisfaction of the users' requirements defined at the beginning of the study. To prove the last point, initial requirements representing pre-requisites of this work have been satisfied by demonstrating that a SPARQL query can be always built to extract the requested information from the knowledge base embodying the narrative. For each of these queries, a specific visualization has been defined to display the data in a user-friendly way. The evaluation by the Dante's expert has confirmed the satisfaction of the requirements and that the ontology is able to express all the requested formalizable knowledge related to the events, their components and relations to Dante's life. This evaluation has also highlighted some key features of the proposed solution, i.e. the possibility to extend the knowledge on each event, using related resources (e.g. Wikipedia pages), digital objects described in external digital libraries (e.g. Europeana) and related images extracted from Wikimedia Commons. Further, the evaluation has also highlighted that, staring from a text, the proposed semi-automatic approach allows to formalize a significant narrative in a relatively short time.

The *primary research question* of this thesis was if an ontology could be created to reasonably represent narratives. The hypotheses thesis were that written natural language of narratives, or oral narrative, could be interpreted and analysed in terms of common and shared interests through digital libraries. Further, these shared interests should be highly generalised and abstracted to an ontological representation. The primary results obtained with the work of this thesis are that (i) a methodological approach can exist that advocates an interpretative analysis of written natural language narratives; more formally, the interpretative analysis can be caught by knowledge engineering and the process of ontological modelling as an act of ontology engineering; (ii) an ontological representation of the narrative has been proposed, which can also be interesting in applications of digital libraries; this uses the CRM ISO standard as reference ontology; (iii) a semi-automatic tool has been built to populate the ontology; (iv) a Web application has been designed and implemented to visualize the knowledge in a simple form, endowed with tables and charts. The validity of the practical application of the presented methodological approach is that the principal common levels of the structures of narrative can be brought to a shared and abstract ontological representation. Even though the CRM primarily describes a material discourse and originates from the museum context, it (and its extension CRMinf) has been found adequate as support to an ontological formalisation of narratives. Further, this outcome can be accessed through queries to be embedded into a Digital Library.

8.2 Future Work

The computational modelling of narratives is an open issue, on which narratologists, linguists, cognitivists, philosophers are still discussing. Nevertheless, the state of the art in narratives, i.e the basic concepts of fabula, narration, plot, events and so on, is solid enough to allow building and successfully evaluating an ontology for narratives. However, this problem requires further investigation beyond this thesis. Some fundamental notions are still missing in the ontology, above all that of character or functional role, which plays a prominent role in narratology. Moreover, it seems inevitable to introduce classes of narratives, each focusing on a specific kind of discourse. In the context of Digital Libraries, besides biography considered in this thesis, the narration of major cultural or historical events seems to be very central. Also the ontology will benefit from this categorization, since it offers the possibility of introducing *ad hoc* classes and properties tailored on the narrative type. As the vocabulary gets richer and richer, also the formalization needs to be enriched with the axioms capturing the semantics of the newly introduced terms. In this respect, the work done in the context of this thesis can be considered as foundational, having laid down the basic elements of a mathematical specification. To this end, a collaboration was started with the Special Interest Group of the CIDOC CRM Ontology targeted at the development of an extension of the CRM for the representation of narratives, which will be proposed to communities working with Digital Libraries. This is due to the fact that the notion of narrative seems to be pervasive in the Cultural Heritage domain, as the work of the humanist revolves around narrative or can often be accounted for in terms of a narrative. But this happens to be true in science as well, where scientific communication is rapidly moving away from the traditional paper, towards complex digital objects including also experimental data, software and a general description of how all these things are tied together. As such, this new form of scientific paper gets very close to a narrative. In order to disseminate and promote this extension, the resulting ontology will be developed and proposed as a domain ontology in the wider context of the CRM standard ontology. Indeed, the CRM has been already used in the Digital Libraries domain and many tools [1] have been developed on the top of it. This situation should favour the adoption of this extension too. The methodology and the tools presented in this thesis are going to be disseminated as Open Source software for research aims starting with a community of scholars in the context of an Italian national research project[2]. As future work, a user test to evaluate the usability and accessibility[3] of the web application that allows to visualize the collected knowledge was planned. Finally, the narrative created as case study in this thesis, i.e. the biography of Dante Alighieri, will be published on the DanteSources Web site[4] to increase the visibility of the work done for this thesis. DanteSources[5] collects and reports Dante's primary sources[6] using Semantic Web technologies and in the last year DanteSources has had 4,956 unique visitors, for a total of 30,501 visited pages.

[1] e.g. http://new.cidoc-crm.org/mapping-tools

[2] http://perunaenciclopediadantescadigitale.eu

[3] https://www.w3.org/WAI/intro/usable

[4] `http://dantesources.org`

[5] DanteSources was the winner of the Digital Humanities Awards 2015 in the category "Best DH Tool or Suite of Tools", `http://dhawards.org/dhawards2015/`

[6] i.e. the works of other authors that Dante cites in his texts.

APPENDIX *A*

RDF Schema in OWL format

CLASSES

CRM Resources

```
Resource crmResource_Entity = model.createResource(Vocabulary.crm + "E1_Entity");
Resource crmResource_ConceptualObject = model.createResource(Vocabulary.crm +
"E28_Conceptual_Object");
Resource crmResource_SymbolicObject = model.createResource(Vocabulary.crm +
"E90_Symbolic_Object");
Resource crmResource_InformationObject = model.createResource(Vocabulary.crm +
"E73_Information_Object");
Resource crmResource_PropositionalObject = model.createResource(Vocabulary.crm
+ "E89_Propositional_Object");
Resource crmResource_Appellation = model.createResource(Vocabulary.crm +
"E41_Appellation");
Resource crmResource_Event = model.createResource(Vocabulary.crm + "E5_Event");
Resource crmResource_Activity = model.createResource(Vocabulary.crm + "E7_Activity");
Resource crmResource_Creation = model.createResource(Vocabulary.crm + "E65_Creation");
Resource crmResource_Actor = model.createResource(Vocabulary.crm + "E39_Actor");
Resource crmResource_Person = model.createResource(Vocabulary.crm + "E21_Person");
Resource crmResource_Place = model.createResource(Vocabulary.crm + "E53_Place");
Resource crmResource_PhysicalThing = model.createResource(Vocabulary.crm +
"E18_Physical_Thing");
Resource crmResource_PhysicalObject = model.createResource(Vocabulary.crm +
"E19_Physical_Object");
Resource crmResource_PersistentItem = model.createResource(Vocabulary.crm +
```

"E77_Persistent_Item");
Resource crmResource_TimeSpan = model.createResource(Vocabulary.crm + "E52_Time_Span");

CRMINF Resources
Resource crminfResource_ObservableEntity = model.createResource(Vocabulary.crminf + "S15_Observable_Entity");
Resource crminfResource_Observation = model.createResource(Vocabulary.crminf + "S4_Observation");
Resource crminfResource_Belief = model.createResource(Vocabulary.crminf + "I2_Belief");
Resource crminfResource_InferenceMaking = model.createResource(Vocabulary.crminf + "I5_Inference_Making");
Resource crminfResource_PropositionSet = model.createResource(Vocabulary.crminf + "I4_Proposition_Set");

NAR Resources
Resource narResource_ActorWithRole = model.createResource(Vocabulary.nar + "ActorWithRole");
Resource narResource_Role = model.createResource(Vocabulary.nar + "Role");

OTHER Resources
Resource rdf_Statement = model.createResource(Vocabulary.rdf + "Statement");
Resource efrbrooResource_ExpressionFragment = model.createResource(Vocabulary.efrbroo + "ExpressionFragment");
Resource dctypesResource_ContentAsText = model.createProperty(Vocabulary.dctypes, "ContentAsText");
Resource dctypesResource_Text = model.createProperty(Vocabulary.dctypes, "Text");

PROPERTIES
rdf:subject
Property rdfProperty_subject = model.createProperty(Vocabulary.rdf, "subject");
model.add(rdfProperty_subject, RDFS.domain, crmResource_InformationObject);
model.add(rdfProperty_subject, RDFS.range, crmResource_Entity);

rdf:predicate
Property rdfProperty_predicate = model.createProperty(Vocabulary.rdf, "predicate");
model.add(rdfProperty_predicate, RDFS.domain, crmResource_InformationObject);
model.add(rdfProperty_predicate, RDFS.range, crmResource_Entity);

rdf:object
Property rdfProperty_object = model.createProperty(Vocabulary.rdf, "object");
model.add(rdfProperty_object, RDFS.domain, efrbrooResource_ExpressionFragment);
model.add(rdfProperty_object, RDFS.range, crmResource_Entity);

dc:format
Property dcProperty_format = model.createProperty(Vocabulary.dc, "format");

```
model.add(dcProperty_format, RDFS.domain, efrbrooResource_ExpressionFragment);
model.add(dcProperty_format, RDFS.range, XSD.xstring);

    cnt:chars
    Property cntProperty_chars = model.createProperty(Vocabulary.cnt, "chars");
model.add(cntProperty_chars, RDFS.domain, efrbrooResource_ExpressionFragment);
model.add(cntProperty_chars, RDFS.range, XSD.xstring);

    crm:hasNote
    Property crmProperty_hasNote = model.createProperty(Vocabulary.crm, "P3_has_note");
model.add(crmProperty_hasNote, RDFS.domain, crmResource_Entity);
model.add(crmProperty_hasNote, RDFS.range, XSD.xstring);

    crm:isComposedOf
    Property crmProperty_isComposedOf = model.createProperty(Vocabulary.crm,
"P106_is_composed_of");
model.add(crmProperty_isComposedOf, RDFS.domain, crmResource_SymbolicObject);
model.add(crmProperty_isComposedOf, RDFS.range, crmResource_SymbolicObject);

    crm:isAbout
    Property crmProperty_isAbout = model.createProperty(Vocabulary.crm, "P129_is_about");
model.add(crmProperty_isAbout, RDFS.domain, efrbrooResource_ExpressionFragment);
model.add(crmProperty_isAbout, RDFS.range, crmResource_InformationObject);

    crm:isIdentifiedBy
    Property crmProperty_isIdentifiedBy = model.createProperty(Vocabulary.crm,
"P149_is_identified_by");
model.add(crmProperty_isIdentifiedBy, RDFS.domain, crmResource_Entity);
model.add(crmProperty_isIdentifiedBy, RDFS.range, crmResource_Appellation);

    crm:beginningIsQualifiedBy
    Property crmProperty_beginningIsQualifiedBy = model.createProperty(Vocabulary.crm,
"P79_beginning_is_qualified_by");
model.add(crmProperty_beginningIsQualifiedBy, RDFS.domain, crmResource_TimeSpan);
model.add(crmProperty_beginningIsQualifiedBy, RDFS.range, XSD.xstring);

    crm:endIsQualifiedBy
    Property crmProperty_endIsQualifiedBy = model.createProperty(Vocabulary.crm,
"P80_end_is_qualified_by");
model.add(crmProperty_endIsQualifiedBy, RDFS.domain, crmResource_TimeSpan);
model.add(crmProperty_endIsQualifiedBy, RDFS.range, XSD.xstring);

    crm:carriedOutBy
    Property crmProperty_carriedOutBy = model.createProperty(Vocabulary.crm,
"P14_carried_out_by");
model.add(crmProperty_carriedOutBy, RDFS.domain, crmResource_Activity);
```

model.add(crmProperty_carriedOutBy, RDFS.range, crmResource_Actor);

crm:hasCreated
Property crmProperty_hasCreated = model.createProperty(Vocabulary.crm,
"P94_has_created");
model.add(crmProperty_hasCreated, RDFS.domain, crmResource_Creation);
model.add(crmProperty_hasCreated, RDFS.range, crmResource_ConceptualObject);

crm:tookPlaceAt
Property crmProperty_tookPlaceAt = model.createProperty(Vocabulary.crm,
"P7_took_place_at");
model.add(crmProperty_tookPlaceAt, RDFS.domain, crmResource_Event);
model.add(crmProperty_tookPlaceAt, RDFS.range, crmResource_Place);

crm:tookPlaceOnOrWithin
Property crmProperty_tookPlaceOnOrWithin = model.createProperty(Vocabulary.crm,
"P8_took_place_on_or_within");
model.add(crmProperty_tookPlaceOnOrWithin, RDFS.domain, crmResource_Event);
model.add(crmProperty_tookPlaceOnOrWithin, RDFS.range, crmResource_PhysicalThing);

crm:occurredInThePresenceOf
Property crmProperty_occurredInThePresenceOf = model.createProperty(Vocabulary.crm,
"P12_occurred_in_the_presence_of");
model.add(crmProperty_occurredInThePresenceOf, RDFS.domain, crmResource_Event);
model.add(crmProperty_occurredInThePresenceOf, RDFS.range, crmResource_PersistentItem);

crm:hasTimeSpan
Property crmProperty_hasTimeSpan = model.createProperty(Vocabulary.crm,
"P4_has_time-span");
model.add(crmProperty_hasTimeSpan, RDFS.domain, crmResource_Event);
model.add(crmProperty_hasTimeSpan, RDFS.range, crmResource_TimeSpan);

crm:represents
Property crmProperty_represents = model.createProperty(Vocabulary.crm,
"P138_represents");
model.add(crmProperty_represents, RDFS.domain, crmResource_Event);
model.add(crmProperty_represents, RDFS.range, crmResource_InformationObject);

crm:consistsOf
Property crmProperty_consistsOf = model.createProperty(Vocabulary.crm,
"P9_consists_of");
model.add(crmProperty_consistsOf, RDFS.domain, crmResource_Event);
model.add(crmProperty_consistsOf, RDFS.range, crmResource_Event);

crminf:observed
Property crminfProperty_observed = model.createProperty(Vocabulary.crm,

```
"O8_observed");
model.add(crminfProperty_observed, RDFS.domain, crminfResource_ObservableEntity);
model.add(crminfProperty_observed, RDFS.range, crminfResource_Observation);

    crminf:observedValue
    Property crminfProperty_observedValue = model.createProperty(Vocabulary.crm,
"O16_observed_value");
model.add(crminfProperty_observedValue, RDFS.domain, crminfResource_Observation);
model.add(crminfProperty_observedValue, RDFS.range, crminfResource_PropositionSet);

    crminf:isSubjectOf
    Property crminfProperty_isSubjectOf = model.createProperty(Vocabulary.crm,
"J4_is_subject_of");
model.add(crminfProperty_isSubjectOf, RDFS.domain, crminfResource_PropositionSet);
model.add(crminfProperty_isSubjectOf, RDFS.range, crminfResource_Belief);

    crminf:wasPremiseFor
    Property crminfProperty_wasPremiseFor = model.createProperty(Vocabulary.crm,
"J1_was_premise_for");
model.add(crminfProperty_wasPremiseFor, RDFS.domain, crminfResource_Belief);
model.add(crminfProperty_wasPremiseFor, RDFS.range, crminfResource_InferenceMaking);

    crminf:concludedThat
    Property crminfProperty_concludedThat = model.createProperty(Vocabulary.crm,
"J2_concluded_that");
model.add(crminfProperty_concludedThat, RDFS.domain, crminfResource_InferenceMaking);
model.add(crminfProperty_concludedThat, RDFS.range, crminfResource_Belief);

    crminf:that
    Property crminfProperty_that = model.createProperty(Vocabulary.crm, "J4_that");
model.add(crminfProperty_that, RDFS.domain, crminfResource_Belief);
model.add(crminfProperty_that, RDFS.range, crminfResource_PropositionSet);

    nar:hadParticipant
    Property narProperty_hadParticipant = model.createProperty(Vocabulary.nar, "had-
Participant");
model.add(narProperty_hadParticipant, RDFS.domain, crmResource_Event);
model.add(narProperty_hadParticipant, RDFS.range, narResource_ActorWithRole);

    nar:subject
    Property narProperty_subject = model.createProperty(Vocabulary.nar, "subject");
model.add(narProperty_subject, RDFS.domain, narResource_ActorWithRole);
model.add(narProperty_subject, RDFS.range, crmResource_Actor);

    nar:role
    Property narProperty_role = model.createProperty(Vocabulary.nar, "role");
```

model.add(narProperty_role, RDFS.domain, narResource_ActorWithRole);
model.add(narProperty_role, RDFS.range, XSD.xstring);

nar:hasSource
Property narProperty_hasSource = model.createProperty(Vocabulary.nar, "hasSource");
model.add(narProperty_hasSource, RDFS.domain, crmResource_InformationObject);
model.add(narProperty_hasSource, RDFS.domain, crmResource_InformationObject);

nar:hasBibliographicReference
Property narProperty_hasBibliographicReference = model.createProperty(Vocabulary.nar, "hasBibliographicReference");
model.add(narProperty_hasBibliographicReference, RDFS.domain, crmResource_InformationObject);
model.add(narProperty_hasBibliographicReference, RDFS.domain, crmResource_InformationObject);

nar:hasTextFragment
Property narProperty_hasTextFragment = model.createProperty(Vocabulary.nar, "hasTextFragment");
model.add(narProperty_hasTextFragment, RDFS.domain, crmResource_InformationObject);
model.add(narProperty_hasTextFragment, RDFS.domain, efrbrooResource_ExpressionFragment);

nar:hasReference
Property narProperty_hasReference = model.createProperty(Vocabulary.nar, "hasReference");
model.add(narProperty_hasReference, RDFS.domain, crmResource_InformationObject);
model.add(narProperty_hasReference, RDFS.domain, XSD.xstring);

nar:causallyDependsOn
Property narProperty_causallyDependsOn = model.createProperty(Vocabulary.nar, "causallyDependsOn");
model.add(narProperty_causallyDependsOn, RDFS.domain, crmResource_Event);
model.add(narProperty_causallyDependsOn, RDFS.range, crmResource_Event);

nar:numBook
Property narProperty_numBook = model.createProperty(Vocabulary.nar, "numBook");
model.add(narProperty_numBook, RDFS.domain, crmResource_InformationObject);
model.add(narProperty_numBook, RDFS.range, XSD.integer);

nar:numChapter
Property narProperty_numChapter = model.createProperty(Vocabulary.nar, "numChapter");
model.add(narProperty_numChapter, RDFS.domain, crmResource_InformationObject);
model.add(narProperty_numChapter, RDFS.range, XSD.integer);

Bibliography

[1] Maristella Agosti, Marta Manfioletti, Nicola Orio, and Chiara Ponchia. Enhancing end user access to cultural heritage systems: Tailored narratives and human-centered computing. In *New Trends in Image Analysis and Processing–ICIAP 2013*, pages 278–287. Springer, 2013.

[2] Khurshid Ahmad and Lee Gillam. Automatic ontology extraction from unstructured texts. In *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*, pages 1330–1346. Springer, 2005.

[3] Harith Alani, Sanghee Kim, David E Millard, Mark J Weal, Wendy Hall, Paul H Lewis, and Nigel R Shadbolt. Automatic ontology-based knowledge extraction from web documents. *Intelligent Systems, IEEE*, 18(1):14–21, 2003.

[4] James F Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[5] James F Allen. Towards a general theory of action and time. *Artificial intelligence*, 23(2):123–154, 1984.

[6] Grigoris Antoniou and Frank Van Harmelen. Web ontology language: Owl. In *Handbook on ontologies*, pages 67–92. Springer, 2004.

[7] Aristotele. *Poetica*. Laterza, 1998.

[8] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[9] Mieke Bal. *Narratology: Introduction to the theory of narrative*. University of Toronto Press, 1997.

[10] Sean Bechhofer. Owl: Web ontology language. In *Encyclopedia of Database Systems*, pages 2008–2009. Springer, 2009.

[11] David Beckett. The design and implementation of the redland rdf application framework. *Computer Networks*, 39(5):577–588, 2002.

[12] Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[13] Floris Jurriaan Bex and Bart Verheij. Story schemes for argumentation about the facts of a crime. In *AAAI Fall Symposium: Computational Models of Narrative*, volume 10, page 04. Citeseer, 2010.

[14] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pages 205–227, 2009.

[15] Christian Blaschke and Alfonso Valencia. Automatic ontology construction from the literature. *Genome informatics*, 13:201–213, 2002.

[16] N Borstw. Construction of engineering ontologies. *Enschede: University of Twente*, 1997.

[17] Johan Bos. Wide-coverage semantic analysis with boxer. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, pages 277–286. Association for Computational Linguistics, 2008.

[18] Dan Brickley and Ramanathan V Guha. Resource description framework (rdf) schema specification 1.0: W3c candidate recommendation 27 march 2000. 2000.

[19] Selmer Bringsjord and David Ferrucci. *Artificial intelligence and literary creativity: Inside the mind of brutus, a storytelling machine*. Psychology Press, 1999.

[20] Paul Buitelaar, Daniel Olejnik, and Michael Sintek. A protege plug-in for ontology extraction from text based on linguistic analysis. In *The Semantic Web: Research and Applications*, pages 31–44. Springer, 2004.

[21] Jorge Cardoso. *Semantic Web Services: Theory, Tools and Applications: Theory, Tools and Applications*. IGI Global, 2007.

[22] Marc Cavazza and David Pizzi. Narratology for interactive storytelling: A critical introduction. In *Technologies for Interactive Digital Storytelling and Entertainment*, pages 72–83. Springer, 2006.

[23] Seymour Chatman. Characters and narrators: Filter, center, slant, and interest-focus. *Poetics Today*, 7(2):189–204, 1986.

[24] Seymour Benjamin Chatman. *Story and discourse: Narrative structure in fiction and film*. Cornell University Press, 1980.

[25] Sudarshan Chawathe, Hector Garcia-Molina, Joachim Hammer, Kelly Ireland, Yannis Papakonstantinou, Jeffrey Ullman, and Jennifer Widom. The tsimmis project: Integration of heterogenous information sources. 1994.

[26] Philipp Cimiano. *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[27] PREMIS Editorial Committee et al. Premis data dictionary for preservation metadata, version 2.0. 2008.

[28] Chris Crawford. *Chris Crawford on interactive storytelling*. New Riders, 2012.

[29] Rossana Damiano and Antonio Lieto. Ontological representations of narratives: a case study on stories and actions. In *OASIcs-OpenAccess Series in Informatics*, volume 32. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2013.

[30] B.A. Davey and H.A. Priestley. *Introduction to lattices and order*. Cambridge, second edition, 2002.

[31] Donald Davidson. *Essays on actions and events: Philosophical essays*, volume 1. Oxford University Press, 2001.

[32] Mike Dean, Guus Schreiber, Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L McGuinness, Peter F Patel-Schneider, and L Andrea Stein. Owl web ontology language reference. *W3C Recommendation February*, 10, 2004.

[33] Martin Doerr. The cidoc conceptual reference module: An ontological approach to semantic interoperability of metadata. *AI Mag.*, 24(3):75–92, September 2003.

[34] Martin Doerr, Chryssoula Bekiari, Patrick LeBoeuf, and Bibliotheque nationale de France. Frbroo, a conceptual model for performing arts. In *2008 Annual Conference of CIDOC*, pages 06–18, 2008.

[35] Martin Doerr, Stefan Gradmann, Steffen Hennicke, Antoine Isaac, Carlo Meghini, and Herbert van de Sompel. The europeana data model (edm). In *World Library and Information Congress: 76th IFLA general conference and assembly*, pages 10–15, 2010.

[36] Martin Doerr, Christian-Emil Ore, and Stephen Stead. The cidoc conceptual reference model: a new standard for knowledge sharing. In *Tutorials, posters, panels and industrial contributions at the 26th international conference on Conceptual modeling–Volume 83*, pages 51–56. Australian Computer Society, Inc., 2007.

[37] William J Doll and Gholamreza Torkzadeh. The measurement of end-user computing satisfaction. *MIS quarterly*, pages 259–274, 1988.

[38] David K Elson and Kathleen R McKeown. A tool for deep semantic encoding of narrative texts. In *Proceedings of the ACL-IJCNLP 2009 Software Demonstrations*, pages 9–12. Association for Computational Linguistics, 2009.

[39] Orri Erling and Ivan Mikhailov. Rdf support in the virtuoso dbms. In *Networked Knowledge-Networked Media*, pages 7–24. Springer, 2009.

[40] David Faure and Claire Nedellec. A corpus-based conceptual clustering method for verb frames and ontology acquisition. In *LREC workshop on adapting lexical and corpus resources to sublanguages and applications*, volume 707, page 30, 1998.

[41] David Faure, Claire Nedellec, and Celine Rouveirol. Acquisition of semantic knowledge using machine learning methods: The system asium. In *Universite Paris Sud*. Citeseer, 1998.

[42] Edward A Feigenbaum and Pamela McCorduck. *The fifth generation*. Addison-Wesley Pub., 1983.

[43] Kate Fernie, Jillian Griffiths, P Archer, K Chandrinos, A de Polo, M Stevenson, P Clough, P Goodale, M Hall, E Agirre, et al. Paths: Personalising access to cultural heritage spaces. In *Virtual Systems and Multimedia (VSMM), 2012 18th International Conference on*, pages 469–474. IEEE, 2012.

[44] Mark A Finlayson. The story workbench: An extensible semi-automatic text annotation tool. In *Intelligent Narrative Technologies*, 2011.

[45] Mark A Finlayson, Whitman Richards, and Patrick Henry Winston. Computational models of narrative: Review of a workshop. *AI Magazine*, 31(2):97, 2010.

[46] Monika Fludernik. Histories of narrative theory (ii): From structuralism to the present. *A companion to narrative theory*, pages 36–59, 2005.

[47] Blaz Fortuna, Marko Grobelnik, and Dunja Mladenic. *Ontogen: Semi-automatic ontology editor*. Springer, 2007.

[48] Roberto Franzosi. Narrative as data: Linguistic and statistical tools for the quantitative study of historical events. *International review of social history*, 43(S6):81–104, 1998.

[49] Marcello Frixione and Antonio Lieto. Representing concepts in formal ontologies. compositionality vs. typicality effects. *Logic and Logical Philosophy*, 21(4):391–414, 2012.

[50] Gerard E Genette and Jane E Lewin. *Narrative discourse: An essay in method*. Cornell University Press, 1983.

[51] Pablo Gervás, Belén Díaz-Agudo, Federico Peinado, and Raquel Hervás. Story plot generation based on cbr. *Knowledge-Based Systems*, 18(4):235–242, 2005.

[52] Pablo Gervás, Carlos León, and Gonzalo Méndez. Schemas for narrative generation mined from existing descriptions of plot. In *OASIcs-OpenAccess Series in Informatics*, volume 45. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

[53] Günther Goerz, Martin Oischinger, and Bernhard Schiemann. An implementation of the cidoc conceptual reference model (4.2. 4) in owl-dl. In *Proceedings of the 2008 Annual Conference of CIDOC-The Digital Curation of Cultural Heritage*, 2008.

[54] Algirdas Julien Greimas, Daniele McDowell, and Alan R Velie. *Structural semantics: An attempt at a method*. University of Nebraska Press Lincoln, 1983.

[55] Thomas R Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5):907–928, 1995.

[56] Nicola Guarino. Formal ontology and information systems. In *Proceedings of FOIS*, volume 98, pages 81–97, 1998.

[57] Nicola Guarino, Daniel Oberle, and Steffen Staab. What is an ontology? In *Handbook on ontologies*, pages 1–17. Springer, 2009.

[58] T Guber. A translational approach to portable ontologies. *Knowledge Acquisition*, 5(2):199–229, 1993.

[59] Sinem Guven, Mark Podlaseck, and Gopal Pingali. Picasso: Pervasive information chronicling, access, search, and sharing for organizations. In *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*, pages 341–350. IEEE, 2005.

[60] Douglas Alan Harrell Jr. *Theory and technology for computational narrative: An approach to generative and interactive narrative with bases in algebraic semiotics and cognitive linguistics*. PhD thesis, University of California, San Diego, 2007.

[61] David Herman. Narratology as a cognitive science. *image and narrative*, 1(1), 2000.

[62] David Herman. *Basic elements of narrative*. John Wiley and Sons, 2011.

[63] Eero Hyvonen, Joeli Takala, Olli Alm, Tuukka Ruotsalo, and Eetu Makela. Semantic kalevala-accessing cultural contents through semantically annotated stories. In *Proceedings of the Cultural Heritage on the Semantic Web Workshop at the 6th International SemanticWeb Conference (ISWC 2007), Busan, Korea*. Citeseer, 2007.

[64] Giuseppe Indizio. *Problemi di biografia dantesca*. Longo, 2014.

[65] Manfred Jahn. Narratology: A guide to the theory of narrative. *English Department, University of Cologne*, 28, 2005.

[66] Roberto J. Bayardo Jr., William Bohrer, Richard S. Brice, Andrzej Cichocki, Jerry Fowler, Abdelsalam Helal, Vipul Kashyap, Tomasz Ksiezyk, Gale Martin, Marian H. Nodine, Mosfeq Rashid, Marek Rusinkiewicz, Ray Shea, C. Unnikrishnan, Amy Unruh, and Darrell Woelk. Infosleuth: Semantic integration of information in open and dynamic environments (experience paper). In Joan Peckham, editor, *SIGMOD Conference*, pages 195–206. ACM Press, 1997.

[67] Daniel Jurafsky and James H Martin. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition. 2000.

[68] Eoin Kilfeather, John McAuley, Anthony Corns, and Oli McHugh. An ontological application for archaeological narratives. In *Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on*, pages 110–114. IEEE, 2003.

[69] Robert Kowalski and Marek Sergot. A logic-based calculus of events. In *Foundations of knowledge base management*, pages 23–55. Springer, 1989.

[70] Elektra Kypridemou and Loizos Michael. Narrative similarity as common summary. In *OASIcs-OpenAccess Series in Informatics*, volume 32. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2013.

[71] Carl Lagoze and Jane Hunter. The abc ontology and model. *Journal of Digital Information*, 2(2), 2006.

[72] R Raymond Lang. A formal model for simple narratives. 1997.

[73] Egoitz Laparra, German Rigau, Piek Vossen, et al. Mapping wordnet to the kyoto ontology. In *LREC*, pages 2584–2589, 2012.

[74] K Faith Lawrence, Michael O Jewell, Adam Prugel-Bennett, et al. Annotation of heterogenous media using ontomedia. 2006.

[75] Michael Lebowitz. Story-telling as planning and learning. *Poetics*, 14(6):483–502, 1985.

[76] Antonio Lieto and Rossana Damiano. A hybrid representational proposal for narrative concepts: A case study on character roles. In *OASIcs-OpenAccess Series in Informatics*, volume 41. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

[77] Charlotte Linde. Social issues in the understanding of narrative. In *2010 AAAI Fall Symposium Series*, 2010.

[78] Hugo Liu and Push Singh. Makebelieve: Using commonsense knowledge to generate stories. In *AAAI/IAAI*, pages 957–958, 2002.

[79] Vincenzo Lombardo and Rossana Damiano. Narrative annotation and editing of video. In *Interactive Storytelling*, pages 62–73. Springer, 2010.

[80] Vincenzo Lombardo and Rossana Damiano. Semantic annotation of narrative media objects. *Multimedia Tools and Applications*, 59(2):407–439, 2012.

[81] Alexander Maedche. *Ontology learning for the semantic web*, volume 665. Springer Science and Business Media, 2012.

[82] Alexander Maedche and Steffen Staab. Ontology learning. In *Handbook on ontologies*, pages 173–190. Springer, 2004.

[83] Alexander Maedche and Raphael Volz. The ontology extraction and maintenance framework text-to-onto. In *Proc. Workshop on Integrating Data Mining and Knowledge Management, USA*, pages 1–12, 2001.

[84] Inderjeet Mani. Computational modeling of narrative. *Synthesis Lectures on Human Language Technologies*, 5(3):1–142, 2012.

[85] Frank Manola and Eric Miller. RDF Primer. W3C Recommendation, WWW Consortium, February 2004. `http://www.w3.org/TR/rdf-primer/`.

[86] Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, and Alessandro Oltramari. Wonderweb deliverable d18, ontology library (final). *ICT project*, 33052, 2003.

[87] Michael Mateas and Phoebe Sengers. *Narrative intelligence*, volume 46. John Benjamins Publishing, 2003.

[88] Brian McBride. The resource description framework (rdf) and its vocabulary description language rdfs. In *Handbook on ontologies*, pages 51–65. Springer, 2004.

[89] John McCarthy. A basis for a mathematical theory of computation. *Computer programming and formal systems*, 354, 1963.

[90] John McCarthy and Patrick J Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Readings in artificial intelligence*, pages 431–450, 1969.

[91] Deborah L McGuinness, Frank Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(10):2004, 2004.

[92] James R Meehan. Tale-spin, an interactive program that writes stories. In *IJCAI*, pages 91–98, 1977.

[93] Carlo Meghini, Nicolas Spyratos, Tsuyoshi Sugibuchi, and Jitao Yang. A model for digital libraries and its translation to rdf. *Journal on Data Semantics*, 3(2):107–139, 2014.

[94] Carlol Meghini and Martin Doerr. A first-order logic expression of the cidoc conceptual reference model.

[95] Jan Christoph Meister. *Computing action: a narratological approach*, volume 2. Walter de Gruyter, 2003.

[96] Jan Christoph Meister. Narratology., 2012. Synthesis. American Museum of Natural History. Available at http://ncep.amnh.org.

[97] Carolyn Handler Miller. *Digital storytelling: A creator's guide to interactive entertainment*. Taylor & Francis, 2004.

[98] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244, 1990.

[99] George A Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G Thomas. Using a semantic concordance for sense identification. In *Proceedings of the workshop on Human Language Technology*, pages 240–243. Association for Computational Linguistics, 1994.

[100] Rob Miller and Murray Shanahan. Some alternative formulations of the event calculus. In *Computational logic: logic programming and beyond*, pages 452–490. Springer, 2002.

[101] Luc Moreau, Ben Clifford, Juliana Freire, Joe Futrelle, Yolanda Gil, Paul Groth, Natalia Kwasnikowska, Simon Miles, Paolo Missier, Jim Myers, et al. The open provenance model core specification (v1. 1). *Future Generation Computer Systems*, 27(6):743–756, 2011.

[102] Erik T Mueller. *Commonsense Reasoning: An Event Calculus Based Approach*. Morgan Kaufmann, 2014.

[103] Paul Mulholland and Trevor Collins. Using digital narratives to support the collaborative learning and exploration of cultural heritage. In *Database and Expert Systems Applications, 2002. Proceedings. 13th International Workshop on*, pages 527–531. IEEE, 2002.

[104] Paul Mulholland, Trevor Collins, and Zdenek Zdrahal. Story fountain: intelligent support for story research and exploration. In *Proceedings of the 9th international conference on Intelligent user interfaces*, pages 62–69. ACM, 2004.

[105] Roberto Navigli, Paola Velardi, and Aldo Gangemi. Ontology learning and its application to automated terminology translation. *Intelligent Systems, IEEE*, 18(1):22–31, 2003.

[106] James Niehaus, Victoria Romero, David Koelle, Noa Palmon, Bethany Bracken, Jonathan Pfautz, Scott Neal Reilly, and Peter Weyhrauch. A flexible framework for the creation of narrative-centered tools. In *OASIcs-OpenAccess Series in Informatics*, volume 41. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

[107] Ana Oliveira, Francisco Camara Pereira, and Amilcar Cardoso. Automatic reading and learning from text. In *Proceedings of the International Symposium on Artificial Intelligence (ISAI)*. Citeseer, 2001.

[108] Peter F Patel-Schneider, Patrick Hayes, Ian Horrocks, et al. Owl web ontology language semantics and abstract syntax. *W3C recommendation*, 10, 2004.

[109] Lyn Pemberton. A modular approach to story generation. In *Proceedings of the fourth conference on European chapter of the Association for Computational Linguistics*, pages 217–224. Association for Computational Linguistics, 1989.

[110] Rafael PÉrez Ý PÉrez and Mike Sharples. Mexica: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence*, 13(2):119–139, 2001.

[111] Cimiano Philipp and Johanna Volker. Text2onto-a framework for ontology learning and data-driven change discovery. In *Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems-NLDB*, volume 5, pages 15–17, 2005.

[112] Lidia Pivovarova and Elena Yagunova. Collocation extraction and text analysis: Different types of collocations and different genres.

[113] David Lynton Poole, Alan K Mackworth, and Randy Goebel. *Computational intelligence: a logical approach*, volume 79. Oxford University Press New York, 1998.

[114] Vladimir Propp. *Morphology of the Folktale*, volume 9. University of Texas Press, 1973.

[115] Yves Raimond and Samer Abdallah. The event ontology. Technical report, Technical report, 2007. http://motools. sourceforge. net/event, 2007.

[116] Mark O Riedl and R Michael Young. Narrative planning: balancing plot and character. *Journal of Artificial Intelligence Research*, 39(1):217–268, 2010.

[117] McKee Robert. Story. substance, structure, style, and the principles of screenwriting, 1997.

[118] Stuart Russell, Peter Norvig, and Artificial Intelligence. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25:27, 1995.

[119] Ansgar Scherp, Thomas Franz, Carsten Saathoff, and Steffen Staab. F–a model of events based on the foundational ontology dolce+ dns ultralight. In *Proceedings of the fifth international conference on Knowledge capture*, pages 137–144. ACM, 2009.

[120] Ryan Shaw, Raphael Troncy, and Lynda Hardman. Lode: Linking open descriptions of events. In *The Semantic Web*, pages 153–167. Springer, 2009.

[121] Victor Shklovsky. Art as technique. *Russian formalist criticism: Four essays*, 3, 1965.

[122] Push Singh et al. The public acquisition of commonsense knowledge. In *Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*, 2002.

[123] Dezhao Song, Christopher G Chute, and Cui Tao. Semantator: annotating clinical narratives with semantic web ontologies. *AMIA Summits on Translational Science Proceedings*, 2012:20, 2012.

[124] Eddie Soulier. *Le storytelling: concepts, outils et applications*. Hermes Science: Lavoisier, 2006.

[125] John F Sowa. Conceptual graphs. *Foundations of Artificial Intelligence*, 3:213–237, 2008.

[126] Steffen Staab and Rudi Studer. *Handbook on ontologies*. Springer Science & Business Media, 2013.

[127] Tzvetan Todorov. *Grammaire du décaméron*. Mouton The Hague, 1969.

[128] Scott R Turner. *The creative process: A computer model of storytelling and creativity*. Psychology Press, 1994.

[129] Jan Uhlir and Martin Falc. Annotating narratives using ontologies and conceptual graphs. In *Database and Expert Systems Applications, 2004. Proceedings. 15th International Workshop on*, pages 84–88. IEEE, 2004.

[130] Willem Robert Van Hage, Veronique Malaise, Roxane Segers, Laura Hollink, and Guus Schreiber. Design and use of the simple event model (sem). *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(2):128–136, 2011.

[131] Frank Van Harmelen, Vladimir Lifschitz, and Bruce Porter. *Handbook of knowledge representation*, volume 1. Elsevier, 2008.

[132] Johanna Volker. *Learning expressive ontologies*, volume 2. IOS Press, 2009.

[133] Johanna Volker, Pascal Hitzler, and Philipp Cimiano. Acquisition of owl dl axioms from lexical resources. In *The Semantic Web: Research and Applications*, pages 670–685. Springer, 2007.

[134] Utz Westermann and Ramesh Jain. A generic event model for event-centric multimedia data management in echronicle applications. In *Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on*, pages x106–x106. IEEE, 2006.

[135] Gio Wiederhold. Mediators in the architecture of future information systems. *Computer*, 25(3):38–49, 1992.

[136] Geoffrey Williams. Mediating between lexis and texts: collocational networks in specialised corpora. *ASp. la revue du GERAS*, (31-33):63–76, 2001.

[137] Annika Wolff, Paul Mulholland, and Trevor Collins. Storyspace: a story-driven approach for creating museum narratives. In *Proceedings of the 23rd ACM conference on Hypertext and social media*, pages 89–98. ACM, 2012.

[138] Gian Piero Zarri. Representing and managing narratives in a computer-suitable form. In *AAAI Fall Symposium: Computational Models of Narrative*, 2010.