Università degli Studi di Udine

Dipartimento di Scienze Matematiche, Informatiche e Fisiche

PhD course in Computer Science, Mathematics and Physics

Ciclo XXXII

Ph.D. Thesis

Visual approaches to knowledge organization and contextual exploration

Candidate:
Marco Corbatto

Supervisor:
Antonina Dattolo

2020

Author's e-mail: marco.corbatto@uniud.it; marco.corbatto@gmail.com

Author's address:

Dipartimento di Scienze Matematiche, Informatiche e Fisiche
Università degli Studi di Udine
Via delle Scienze, 206
33100 Udine
Italia

Abstract

This thesis explores possible visual approaches for the representation of semantic structures, such as *zz-structures*. Some holistic visual representations of complex domains have been investigated through the proposal of new views - the so-called zz-views - that allow both to make visible the interconnections between elements and to support a contextual and multilevel exploration of knowledge. The potential of this approach has been examined in the context of two case studies that have led to the creation of two Web applications.

The first domain of study regarded the visual representation, analysis and management of scientific bibliographies. In this context, we modeled a Web application, we called *VisualBib*, to support researchers in building, refining, analyzing and sharing bibliographies. We adopted a multi-faceted approach integrating features that are typical of three different classes of tools: bibliography visual analysis systems, bibliographic citation indexes and personal research assistants. The evaluation studies carried out on a first prototype highlighted the positive impact of our visual model and encouraged us to improve it and develop further visual analysis features we incorporated in the version 3.0 of the application.

The second case study concerned the modeling and development of a multimedia catalog of Web and mobile applications. The objective was to provide an overview of a significant number of tools that can help teachers in the implementation of active learning approaches supported by technology and in the design of Teaching and Learning Activities (TLAs). We analyzed and documented 281 applications, preparing for each of them a detailed multilingual card and a video-presentation, organizing all the material in an original purpose-based taxonomy, visually represented through a browsable holistic view. The catalog, we called *AppInventory*, provides contextual exploration mechanisms based on zz-structures, collects user contributions and evaluations about the apps and offers visual analysis tools for the comparison of the applications data and user evaluations. The results of two user studies carried out on groups of teachers and students shown a very positive impact of our proposal in term of graphical layout, semantic structure, navigation mechanisms and usability, also in comparison with two similar catalogs.

Acknowledgments

This thesis is the result of three very intense years of research and development that have given me the opportunity to enrich myself also by participating in several collaborative projects, conferences and by interacting with colleagues. I would like to thank in primis my supervisor, professor Antonina Dattolo, for this wonderful experience, for everything she taught me, for her countless suggestions and for the always positive and cheerful atmosphere of collaboration within the Sasweb laboratory that quickly made it possible to overcome every difficulty encountered.

I would like to thank Meshna Koren and Dave Santucci for confirming the Elsevier/Scopus interest in our VisualBib project and for the technical support; Mirco Ianese for his contribution in the development of some features of the VisualBib platform; Marco Angelini for his contribution in terms of ideas that led to the development of version 3.0; Alessandro Iop and Martina Urizio for their help in organizing and revising the materials of the AppInventory catalog.

A thank you to all students that enthusiastically participated to the AppInventory project and have contributed in different ways to the birth of this catalog. We reported all their names on a dedicated page `http://appinventory.uniud.it/en/about-us/` of the AppInventory site, hoping not to have forgotten anyone.

I would like also thank Professors Alessandra Pallavicini and Giordano Vintaloro for their support in proofreading of part of this work. Finally, a big thank to my family, my wife Donatella and my son Giacomo who have never stopped encouraging and supporting me during this journey and to my elderly parents to whom this work is dedicated.

Contents

List of Figures

List of Tables

Introduction

Nowadays, the diffusion of Web applications for real-time access to large amounts of data or integrated services is widespread. Despite the evolution of interfaces and technical solutions to offer a fluid and intuitive access to information in a domain, the original and effective representation and contextual exploration of semantic structures is still an open field of research. The main objectives of this work are to:

- explore the potential of zz-structures, introduced for the first time by Theodor Holm Nelson in 1998 [87], starting from the ideas developed in the Xanadu project [88], for the representation and the browsing on data and knowledge;

- apply on two different case studies that led to the creation of real systems inspired by Nelson's ideas.

Among the main features of zz-structures is the wide freedom they offer to semantically link single information, enclosed in cells, across multiple dimensions. Each dimension represents a way to semantically link the cells: if we think to cells representing paintings, a dimension could be the *authorship* that connects all works of each painter, another dimension could be *collocation* which links all the paintings exhibited in a same museum, and so on. What we get is a potentially very complex multidimensional space that allows users to navigate in linear paths along a certain dimension and then move, changing perspective, along another dimension. A metaphor, already found in literature [41], is that of a subway which allows a traveller to reach a series of nodes in a city by moving along interconnection lines marked with colors. Each color represents a type of semantic link between the nodes, we could think that there is a yellow line that connects the administrative centres of a city, a red line that connects the artistic ones, a blue line that allows to move between points of commercial interest and so on. It is clear that a node can be found at the intersection of several lines, giving the traveller the opportunity to change perspective on his journey, deciding to explore the nearby nodes according to a different criterion. The device proposed by Nelson for the contextual exploration of a certain node's surroundings is that of zz-views which offer one or more "sustainable" ways of moving in a structure that is often too complex to be considered as a whole. One idea that zz-views convey is *the visibility of interconnections* [86]: in the hypermedia space of the Web, we are all used to moving through links; the problem is that on every page there are many links, they are generally not labeled semantically, except through their anchor, and they do not commonly offer linear approaches to exploration. Moreover, they are normally unidirectional, leaving to the browser the task of memorizing the road travelled in order to recover a significant starting point. The user is imposed the cognitive load of the construction and maintains a mental map of his movements in the hyper-structure, which the system generally does not provide. The idea of the "visibility" and the bi-directionality of the interconnections provided by the zz-view offers the opportunity to know something about what you will visit even before you do so and, in any case, to be able to meet similar elements according to the current criterion (dimension) of navigation, in addition to returning to the already visited nodes. Another important principle, conveyed in this case by information visualization, is the well-known

Visual Information Seeking Mantra [92]: *Overview first, zoom and filter, then details-on-demand.* Is it possible to combine this idea of overview with that of contextual navigation conveyed by zz-views? In the first case study we tried to offer an overview of an entire zz-structure through views that initially concealed both the details and some of the dimensions. We took care to make visible only the connections between the nodes of the current "active" dimensions and to provide mechanisms of contextual and temporarily highlighting of the set of nearby nodes of a focused one, hiding everything else from the view. In the context of the second application we adopted a particular zz-view that presents an overview, according to a subset of dimensions, offering a semantic zoom mechanism to progressively reveal details and access navigation mechanisms along the other dimensions. More in particular, the two addressed case studies regard:

1. the modeling, the design and the implementation of the VisualBib Web app; it is dedicated to the building, refining, representing, analyzing and sharing of scientific bibliographies;

2. the design and the implementation of the AppInventory Web app, a multimedia catalog of Web 2.0 and mobile applications.

The original contributions of this thesis are:

1. the extending of the existing models of zz-views with three new proposals: *deep-views*, *narrative-views* and *bubble-views* described in Section 1.3;

2. the publication of an interactive visual representation of a comprehensive bibliography about zz-structures, accessible at `http://zzstructure.uniud.it` described in Section 1.4;

3. the modeling and the realization of a Web platform described in Chapter 2, called VisualBib^{TM}, currently in the version 3.0 and accessible at `http://visualbib.uniud.it`, for supporting researchers in the building, refining, representing, analyzing and sharing of scientific bibliographies. VisualBib enables users to import a bibliography in a BibTeX format, enrich it by querying Scopus APIs, detect duplicated names of authors, integrate the bibliography with new papers by realtime metadata retrieval from external indexes and carry out visual analysis tasks. VisualBib inserts and connects new elements in an holistic view of the bibliography called narrative-view which highlights the temporal collocations of papers, the network of collaborations between the authors, the citation network and the semantic relationships between papers given by keywords, tags and subject areas;

4. the modeling and realization of AppInventory described in Chapter 3, a multimedia catalog of 281 Web 2.0 and mobile applications; it is based on an original purpose-based taxonomy and is freely accessible at `http://appinventory.uniud.it`. Each application has been selected, analyzed and documented in multilingual information cards and original video-presentations with the aim to support teachers, students and professionals in finding best tools to carry out specific activities. The platform adopts an innovative approach for exploring the catalog, represented as a 2D space navigable through a semantic zoom mechanism and linear paths along multiple-dimensions. The catalog also collects user contributions and evaluations and offers a visual analysis tool for the comparison of the applications.

Outline

We outline here the structure of this thesis. Figure 1 shows its representation through a zz-structure where the zz-cells are the sections which are interconnected by 6 dimensions labeled by meta-descriptors.

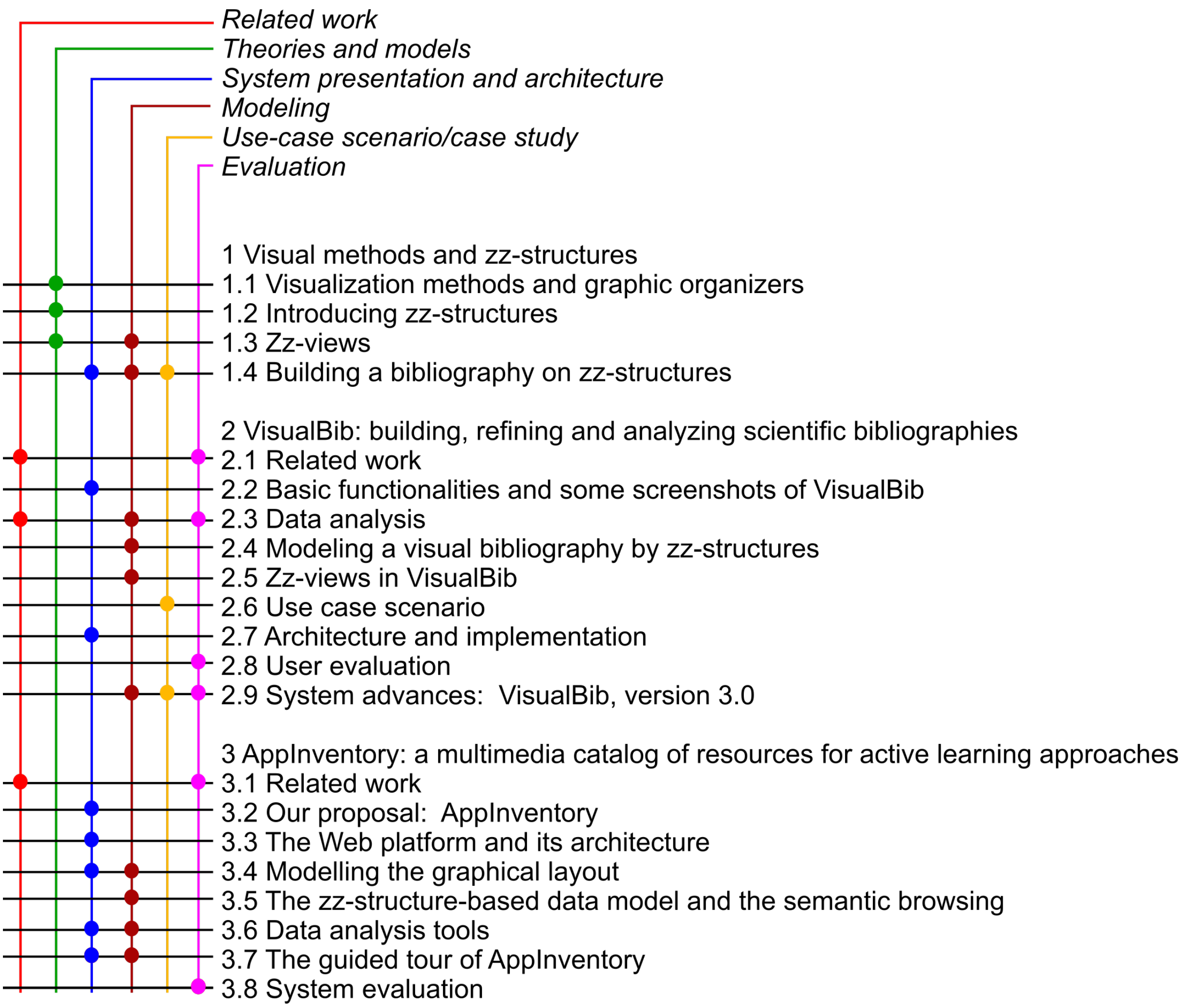

Figure 1: The outline of this thesis in the form of a zz-structure.

Chapter 1 introduces concepts, models and formalizations we apply in the case studies illustrated in Chapters 2 and 3. In Section 1.1 we start presenting some definitions, models, guidelines, classifications, technologies and frameworks regarding visualization methods. The aim is to introduce some general concepts concerning this subject before contextualising them in the model of zz-structures. This model is then introduced and formalized in Section 1.2 while the concept of zz-view is presented in Section 1.3 where a series of relevant zz-views proposed in literature are outlined. In this context, three new original zz-views are introduced and defined for the purpose of their application in the two case studies. Section 1.4 presents a comprehensive bibliography on zz-structures modeled through a zz-structure itself and represented by means of the newly introduced *narrative-view* and *deep-view*. A prototype of an interactive visual representation of the zz-bibliography has been included in the website which indexes all the collected papers. Chapter 2 is dedicated to the VisualBib™project, a Web application for supporting researchers in building, refining, analyzing and sharing scientific bibliographies.

The project, inspired both by the Nelson's zz-structures and by the initial prototype of the interactive visual representation of the zz-bibliography, led to the realization of a complex live system which interfaces with external bibliographic indexes to retrieve real-time metadata and integrate them in a visual environment. The chapter analyzes in detail related work about tools for the visual analysis of bibliographies and some bibliographic indexes about the exposed data and the offered Application Program Interface (API) services. After a presentation of the basic functionalities of the system, we introduce the model of a bibliography, based on a zz-structure, and the zz-views built over it, addressing the topological constraints for the generation of a narrative-view. The chapter continues with the illustration of a use case scenario consisting in the import of a bibliography from a BibTeX archive, its enrichment through the retrieval of extended metadata from Scopus, the detection and merging of duplicate authors' names, the refinement of the bibliography by integrating new references and, finally, its exporting. The system architecture is then illustrated together with some implementation details. The user evaluation section presents two studies carried out in order to estimate the perceived usability of the platform and its effectiveness in performing some research tasks, also in comparison with a traditional bibliographic index platform. The last section presents the new version 3.0 of VisualBib which incorporates, in a completely renovated environment, several features for the visual analysis of bibliographies; the section ends with the presentation of the results of an evaluation study and a demonstration of a visual analysis task carried out on a sample bibliography.
Chapter 3 focuses on the AppInventory project introducing the motivations and the analysis of related work, illustrating the cataloging scheme, the purpose-based taxonomy, the process of building the repository and some statistics about dataset. After the presentation of the system architecture and the main features of the platform, a model of the catalogue based on zz-structure is introduced together with a nested version of the new bubble-view proposal of zz-view, which formally defines the main view of the catalog. Then we present a data analysis tool integrated in the platform and a guided tour of the platform, also modeled through a zz-structure. The chapter ends with the illustration of the two user studies for the evaluation of the system.

Contributions

A considerable part of this work has been created during collaboration with the chief of the SASWEB[1] Research Lab., professor Antonina Dattolo. This section clarifies the author's contribution to each publication and its corresponding chapter within this thesis.

Chapter 2

- Antonina Dattolo and Marco Corbatto. VisualBib: A novel Web app for supporting researchers in the creation, visualization and sharing of bibliographies.
  In: *Knowledge-Based Systems*, vol.182, 2019, 104860, ISSN 0950-7051,
  DOI: `https://www.doi.org/10.1016/j.knosys.2019.07.031`.

- Antonina Dattolo and Marco Corbatto. VisualBib: Narrative Views for Customized Bibliographies. In: *Proceedings of the 22nd International Conference Information*

[1]Semantic Adaptive Social Web Research Lab - Department of Mathematics, Computer Science, and Physics, University of Udine - `http://sasweb.uniud.it`

Visualisation – IV2018, Salerno, Italy. July 10 – 13 2018, IEEE, pp. 133-138, DOI: https://www.doi.org/10.1109/iV.2018.00033.

- Marco Corbatto and Antonina Dattolo. A Web Application for Creating and Sharing Visual Bibliographies. In: *González-Beltrán A., Osborne F., Peroni S., Vahdati S. (eds) Semantics, Analytics, Visualization. SAVE-SD 2017 and SAVE-SD 2018. Lecture Notes in Computer Science*, vol. 10959, 2018, Springer, Cham, pp. 78-94, DOI: https://www.doi.org/10.1007/978-3-030-01379-0, eBook ISBN: 978-3-030-01379-0, Softcover ISBN: 978-3-030-01378-3.

Chapter 3

- Marco Corbatto and Antonina Dattolo, Exploring AppInventory, a visual catalog of applications for assisting teachers and students. In: *Multimedia Tools and Applications*, 2019, ISSN: 1573-7721, DOI: https://www.doi.org/10.1007/s11042-019-08000-6.

- Marco Corbatto and Antonina Dattolo. Organizing and evaluating resources and tools for active learning approaches. In: *GoodTechs '19 Proceedings of the 5th EAI International Conference on Smart Objects and Technologies for Social Good*, Valencia, Spain — September 25 - 27, 2019, DOI: https://www.doi.org/10.1145/3342428.3342668.

- Marco Corbatto and Antonina Dattolo. AppInventory: a Visual Catalogue of Web 2.0 and Mobile Applications for Supporting Teaching and Learning Activities. In: *Proceedings of the 22nd International Conference Information Visualisation – IV2018*, Salerno, Italy, July 10 – 13 2018, IEEE, pp. 530-535, DOI: https://www.doi.org/10.1109/iV.2018.00098.

- Marco Corbatto. Modeling and developing a learning design system based on graphic organizers. In: *Adjunct Publication of the 25th International Conference on User Modeling, Adaptation and Personalization (UMAP 2017)*, Bratislava; Slovakia; July 9-12 2017, pp. 117-118. ISBN: 978-1-4503-5067-9, DOI https://www.doi.org/10.1145/3099023.3099028.

1

Visual methods and zz-structures

This chapter starts presenting some definitions, models, guidelines, classifications, technologies and frameworks regarding the visualization methods. The aim is to introduce some general concepts concerning the information visualization before contextualising them in the model of zz-structures and in particular of the zz-views. In particular Section 1.1 introduces a definition of *Visualization method* and *Graphic organizer*, the *Visual Expression Process*, a model which describes how visualization improves knowledge acquisition, some design recommendations, then presents and compares three classification schemes for visual representations and, finally, some technologies and frameworks to develop and incorporate interactive graphic organizers in modern Web applications.

In the next Section 1.2 we briefly describe zz-structures, in order to provide the theoretical basis for introducing the data models of the two treated case studies. ZigZag(tm) is a registered trademark in the USA for zz-structure-based software of Ted Nelson's Project Xanadu. Zz-strucutures represent a generalized representation for data, a new set of mechanisms for computing, and a different way of linking and organizing information; they provide both data representation and exploring mechanisms.
In zz-structures information is organized in multiple dimensions lists. Through the zz-views the user sees a locally relevant view of the information, irrespective of how complex the structure is, and without losing the ability to navigate all the interconnections. In the present thesis the zz-structures represent the reference model to manage the complex data domains of the two case studies, presented in Chapters 2 and 3, together with the multilevel views and contextual exploration mechanisms. We remind interested readers to refer to general papers [89, 59, 60] or the complete list of papers on this topic [55].

Section 1.3 presents some relevant zz-views proposed in the literature; then we propose, in Subsections 1.3.4- 1.3.6, some new and original zz-views which will be applied in the context of the two considered case studies. Specifically, the so-called *deep-view*, *narrative-view* and *bubble-view*.

Section 1.4 presents a research project on zz-structures [33], started in 2017 and still active, with the aim of collecting, at the best of our knowledge, the complete bibliography about zz-structures. In order to formally model a bibliography, a zz-structure has been adopted and some zz-views defined on it with the aim to publish the zz-bibliography on a website using a visual metaphor based on a narrative-view and multiple deep-views. In addition to this, a traditional access to paper metadata and full text has been implemented.

The original contributions of this thesis presented in this chapter are:

1. the extending of the existing models of zz-views with three new proposals: *deep-views*, *narrative-views* and *bubble-views* described in Section 1.3;

2. the publication of an interactive visual representation of a comprehensive bibliog-

raphy about zz-structures, accessible at `http://zzstructure.uniud.it` described in Section 1.4;

1.1 Visualization methods and graphic organizers

In order to introduce the guidelines for the design of visual representations of the two case studies examined in the Chapters 2 and 3, in this section we present some definitions, models, guidelines and classifications regarding the visualization methods.
The last Subsection 1.1.3 presents some technologies and frameworks for developing and integrating interactive visual representations into Web applications.

Graphic organizers are visualization methods to represent knowledge, concepts, or ideas, and the relationships between them [84].

The word *visualization* often refers to the graphic display of information and knowledge [39]; a more specific definition of a *visualization method* can be found in [77]:

> A *visualization method* is a systematic, rule-based, external, permanent, and graphic representation that depicts information in a way that is conducive to acquiring insights, developing an elaborate understanding, or communicating experiences.

The benefits of visual representations are widely recognized and have been described in a many papers [85], [64], [34] but their careful design is a crucial point to avoid common errors and the possible pitfalls when creating or interpreting visual representations. An interesting analysis and an extended classification of common visual representation pitfalls can be found in [39].

In literature we can also find the alternative term “graphic organizer”; a definition [66] is the following:

> A *graphic organizer* (GO) is a visual and graphic display that depicts the relationships between facts, terms, and or ideas within a learning task.

The term *organizer* leads to interpret a GO not only as a representation method but also as a tool, provided with interactive features to modify, transform and adapt to the users' view of information.

Eppler at al. [63] extend the concept introducing the *Dynagram*:

> An interactive visualization that allows users to collaboratively create, alter or extend it to conduct analyses, explore scenarios, make insights, jointly visible or record experiences, evaluations, and decisions.

This formulation underlines the benefits of such representations in a dynamic and constructivist approach to knowledge representation and management.

In general, exploring information in a complex domain becomes increasingly difficult as the volume of information grows [92]: in this case the availability of an holistic view of data with direct-manipulation graphical capabilities, can make the difference in identifying data relationships, in stimulating insight, in supporting navigation and, at the same time, limiting the perceived information overload.

There are many visual design guidelines but a basic principle, suggested by Shneiderman [92], is known as the *Visual Information Seeking Mantra*: Overview first, zoom and filter, then details-on-demand.

1.1.1 From visual stimuli to knowledge

In order to analyze the process of transforming visual stimuli into high level data interpretation and knowledge production, Rodrigues et al. [90] introduced the *Visual Expression Process* as a scheme, visible in Figure 1.1, which describes how visualization improves knowledge acquisition. The presence of one or multiple pre-attentive stimuli (basically

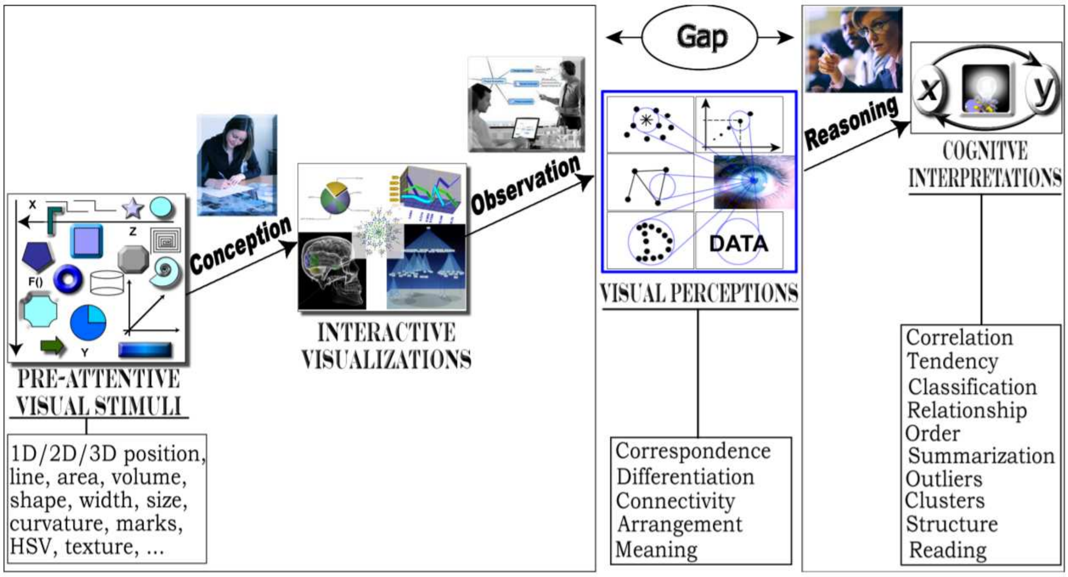

Figure 1.1: The Visual Expression Process. Source: [90].

based on *position*, *shape* and *colors* of visible elements) induces a *visual perception* in the user. This is a recognition phase which highlights the *correspondence* (with respect shape, position, color, to existing mental representations), the *differentiation* (discrimination between graphical items), the *connectivity* (perception of the relationships between the items), the *arrangement* (detection of structural clues of a group of items like continuity, proximity, symmetry) and the *meaning* (matching with interpretations and concepts in long term memory, to detect meaning from correspondence).

In this model, the third and last phases refer to *cognitive interpretation*, which produces deductions, inferences or conclusions from the visual perceptions; the effectiveness of the interpretation will clearly depends both from the knowledge of the domain of the data owned by the user and from the visual representation of the data. In this optic, the presence and opportune organization of visual stimuli and the graphical typology and design of the information assumes a big importance to the overall interpretation and information communication.

In order to extend this model to the dynamic visual representations, we could consider the *movement* as an additional pre-attentive stimulus. In this optic, an interactive visual representation could be interpreted (besides the functionalities provided to user to manipulate data) as a way to alter the parameters of the pre-attentive components in order to stimulate new deductions and interpretations of data. Therefore, giving the user the possibility to interact with the visual representation to show/hide some items, to emphasize relationships, to reorganize the size and the arrangement of the elements, could stimulate pre-attentive mechanism and induce new data inferences and interpretation of the data domain. This interpretation can be applied to animations and graphic transitions too, recognizing them a cognitive function beyond the aesthetic one.

The *Visual Expression Process* model has been recently improved [69] introducing the *time* among the pre-attentive stimuli and redefining the visual perception in two phases: the *attentive selection* (recognition of *visual analytical perceptions*) and the *pattern matching* (recognition of *abstract analytical patterns*) both preceding the cognitive interpretation.

From the perspective of each of the four phases, Rodrigues at al. [69] formulate respective recommendations for the design of a visualization system:

> *Recommendation 1*: visualizations must present features that are potentially pre-attentive in a way that users can interactively redefine each visual stimuli.
>
> *Recommendation 2*: visual perceptions, which are recurrently observed in visualization techniques, must be the bases for design, and evaluation.
>
> *Recommendation 3*: the design of systems shall consider a perceptual perspective, offering users pattern and domain-oriented choices rather than design-oriented choices.
>
> *Recommendation 4*: InfoVis systems must define systematic means to aid the user in recording and accessing the domain knowledge related to the problems at hand.

1.1.2 Classifications

Many classifications of visual representations have been proposed in literature. Lohse et al. [80] proposed eleven categories of visual representations emerged from an experimental study. Participants were asked to evaluate a series of visual representations using ten Likert scales (spatial-nonspatial, nontemporal-temporal, hard-easy to understand, concrete-abstract, continuous-discrete, attractive-unattractive, whole-part emphasized, nonnumeric-numeric, conveys a lot-little information). The emerged categories from their studies were: graphs, tables, graphical tables, time charts, networks, structure diagrams, process diagrams, maps, cartograms, icons and pictures.

In his analysis, Shneiderman [92] proposed a *type by task taxonomy* for information visualization: seven data type have been identified: *1-dimensional data* (sequential list of items), *2-dimensional data* (planar data, maps, for example geographic maps, newspaper layouts, ...), *3-dimensional data* (real-word objects by 3D computer graphic, 3D versions of trees or networks, excluding multi-layers of 2d representations), *temporal* (timeline representations with possible overlapping of items), *multi-dimensional data* (complex representations with multiple attributes items, for example [26]), *tree* (collections of items organized in a hierarchical structure where items and links can have multiple attributes) and *network data* (a generalization of tree structure to model arbitrary relationships between items). All these typologies can be combined to generate more complex representations. Furthermore the following seven typical user's tasks on these representations are classified. *Overview*: gain an overview of the entire collection. *Zoom*: zoom in on items of interest. *Filter*: filter out uninteresting items. *Details-on-demand*: select an item or group and get details when needed. *Relate*: view relations hips among items. *History*: keep a history of actions to support undo, replay, and progressive refinement. *Extract*: allow extraction of sub-collections and of the query parameters.

A significant effort to structure the vast domain of visualization methods has been carried out by Lengler and Eppler [77]. From an initial set of 160 potential visual methods candidates, they reduced it to one hundred applying a fixed set of selection criteria. The resulting set has been analyzed in respect of seven properties (graphic format, typical content type, application context, scope, difficulty of their application, originating discipline, vicinity over overlaps to other visual methods).

The collected data have been finally mapped along five independent dimensions to build a significant classification chart using a periodic table as visual metaphor. The identified dimension are: *groups* (data, information, concept, metaphor, strategy, compound knowledge), *complexity of visualization* (low to high), *point of view* (detail, detail and overview, overview), *type of thinking* (convergent, divergent), *type of representation* (process, structure).

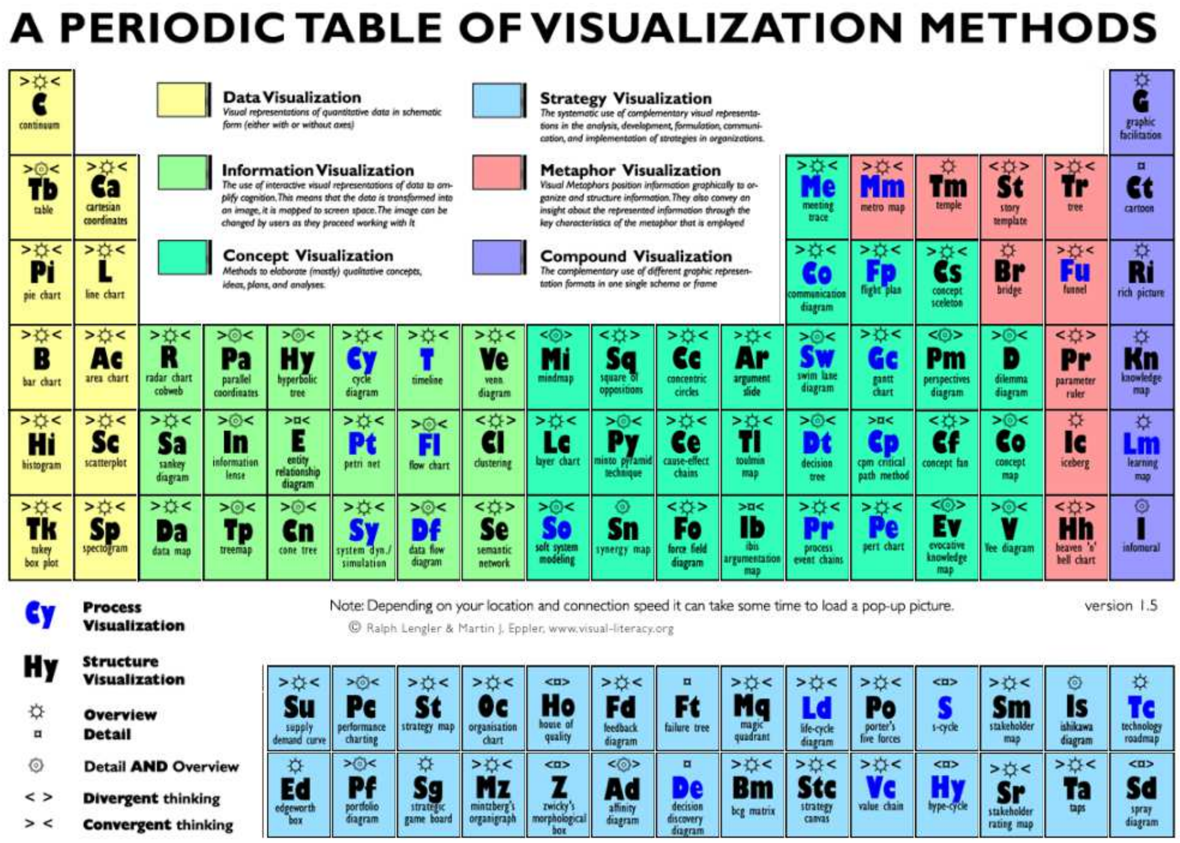

Figure 1.2: A periodic table of visualization methods. Source: [77].

The result chart [28], see Figure 1.2, highlights the five dimensions using a set of pre-attentive stimuli: various background colors for *groups*, vertical position of items for *complexity*, font color for the *type of representation*, special symbols to mark the *point of view* and the *type of thinking*. This table chart, enriched with interactivity (moving the cursor over each element, the complete name of the visualization method will appear together with and a sample), represents an interesting effort to give an holistic view of the realm of visualization methods and to help researches and professionals in considering various alternatives for each visualization requirement.

The heterogeneity of the three classification schemes, does not permit a direct comparison: analyzing the descriptions and the samples diagram presented by the authors we have built a possible mapping between categories of the three schemes, represented in the Sankey diagram of Figure 1.3. On the left are reported the seven data types described by Shneiderman in [92], in the middle the classification introduced by Lohse et al. [80] and on the right the labels of the group dimension in the classification of Lengler and Eppler [77].

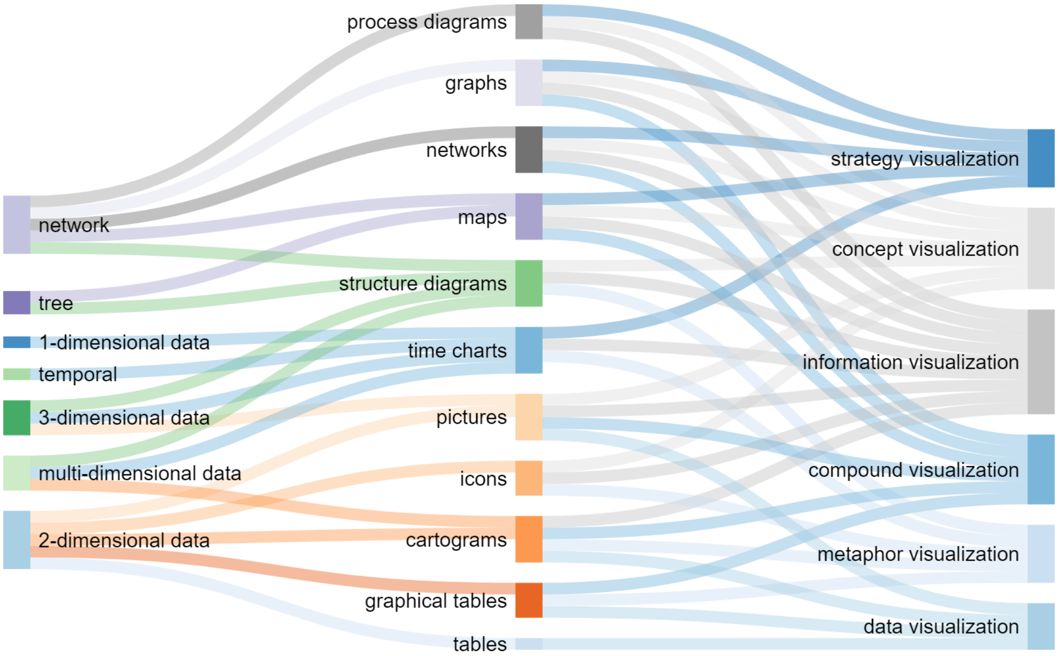

Figure 1.3: A possible mapping of categories in the analyzed classification schemes. On the left the seven data types described by Shneiderman in [92], in the middle the classification introduced by Lohse et al. [80] and on the right the labels of the group dimension in the classification of Lengler and Eppler [77].

1.1.3 Technologies and frameworks

In order to build and incorporate interactive graphic organizers in modern Web applications, it is necessary to adopt technologies based on current Web standards: HTML5 for page structure and contents, CSS for style application, SVG to visualize scalable graphic objects, JavaScript for the application logic and interactivity, and so on. HTML elements and graphics objects in a SVG application are mapped into the document object model (DOM) of the page and can be accessed / manipulated by standard API. These native API set is rich and comprehensive but not particularly expressive and generally provides primitives to manipulate one element at a time. For these reasons application code often results verbose, hard to read and to maintain.

To overcome this problem, many toolkits (such as Processing [25], Flare [29], InfoVis [30], ProtoVis [31]) have been proposed for the development of graphic applications. Each toolkit provides abstract and specialised API or classes for the graphic modelling. While highly facilitating the task of building applications, this additional layer introduces intermediate representations that encapsulate and hide underlying structures preventing the developer to access directly to them, also when advisable. The higher abstraction level brings with it the benefit of an higher efficacy and speed in writing code but introduces some drawbacks too [38]: a computational overhead in execution, the necessity for a developer to learn a new framework without taking advantage of his/her deep knowledge of the Web standards, the lack of integration with already existing efficient standard solutions (for example CSS3 transform and transitions, SVG graphic paths) and, finally, a not easy debugging of applications due to the additional abstraction layer introduced.

An alternative and more agile approach is to integrate the native DOM manipulation primitives by introducing specialised class and functions: JQuery [27] is a successful example of this philosophy, providing a powerful alternative to interact efficiently with the DOM through a declarative selection mechanism and numerous manipulation methods. Nevertheless, it is not the ideal tool for graphic manipulation due to its inability to strictly tie the data with DOM elements in order to create, update or delete them when data changes. Another elegant approach to DOM manipulation is represented by XSLT that provide a declarative method for transforming the object in the page but it has not the flexibility of imperative or object paradigms to face with complex operations.

To build our prototypes for the case studies, we choose D3.js [38] [32], a modular JavaScript library for creating interactive documents with a strong emphasis on Web standards HTML, SVG, and CSS. Similarly to JQuery, D3 provides a powerful DOM selection mechanism based on declarative CSS patterns, a rich library of methods to create complex graphical representations and layouts and to act, with the same syntax, both on single DOM elements and on sets of them.

The idea is to strictly tie data to HTML or SVG elements realizing a so-called *data-driven* approach to DOM manipulation. Through a set of powerful methods, it is relatively easy to create, update and remove elements in a Web page when data change. Transitions and animations can be defined using specific functions that smoothly interpolate, over the time, the style properties of a selected set of elements. D3 provides also many helper modules to define dynamic axes and scales, to parse data files, to interact with asynchronous http requests, to manage and transform data and to support rich visualizations through specialized layouts. An overall view of the D3.js (version 4) framework which highlights its modular structure is visible in Figure 1.4.

Typical steps in a D3's application are:

1. Loading data from an external source in JSON, CSV or others delimited formats. The source can be a local file or the response of an asynchronous call to external APIs. The data are generally loaded in an array of objects and, in order to build complex visualizations (tree layout, pie layout, chord layout, ...), it could be necessary to perform data transformations using specialized layout functions. Of course the loading data step can be repeated in the application's life cycle to retrieve new data as a consequence, for example, of the user's actions on the visual representation.

2. Binding data to a set of DOM elements in order to create the missing elements, to delete those not used anymore and to update the previously mapped. It is also possible to bind event listener functions to any element in order to implement the interactive functionalities of the application.

3. Rendering the data applying specific properties to the attached DOM elements (typically SVG elements), positioning them using scales functions, defining transition to smoothly change the properties from their current values to new ones, easily obtaining animations and dynamic data rendering.

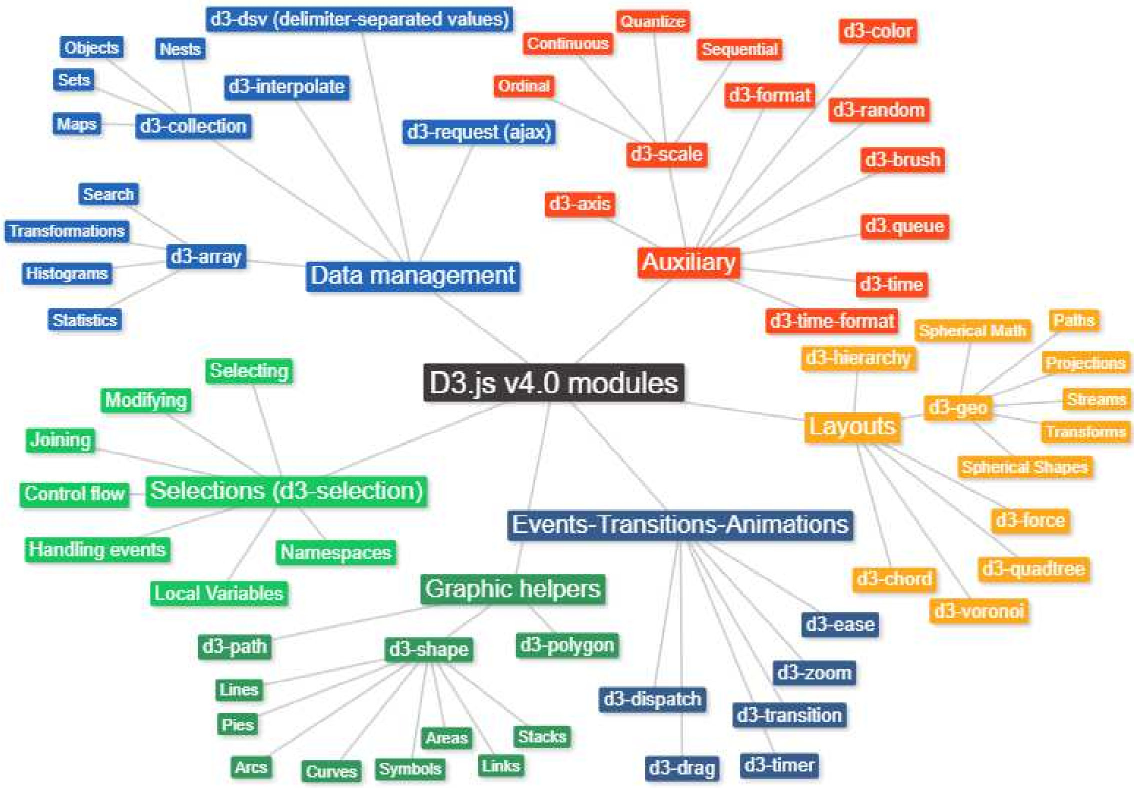

Figure 1.4: A map showing the modular organization of D3 v4.0 library [32]. The d3-names identify the corresponding D3 modules: each provides one or more groups of methods.

The D3's choice of not hiding the underlying DOM structure of the application [43] makes it possible to take advantage of the integrated debuggers generally embedded in modern browsers. These powerful tools help the developers to easily navigate into the DOM, to examine and change markup code and object properties, to dynamically inquiry and modify the style attributes, to debug code by watching variables, inserting breakpoints, analyzing stack, local storage and finally, to monitor the network activity, see Figure 1.5. In summary we chose to adopt D3 as the reference framework for developing all the Web applications presented in this thesis, for a number of reasons:

- the richness of object and methods provided; agile coding thanks to methods chaining;
- its transparency towards the underlying DOM layer;
- the compatibility between browsers;
- the capability to attach data to DOM elements;
- same methods for getter and setter; setters accept *accessor function* as argument;
- easy management of animations and transitions.

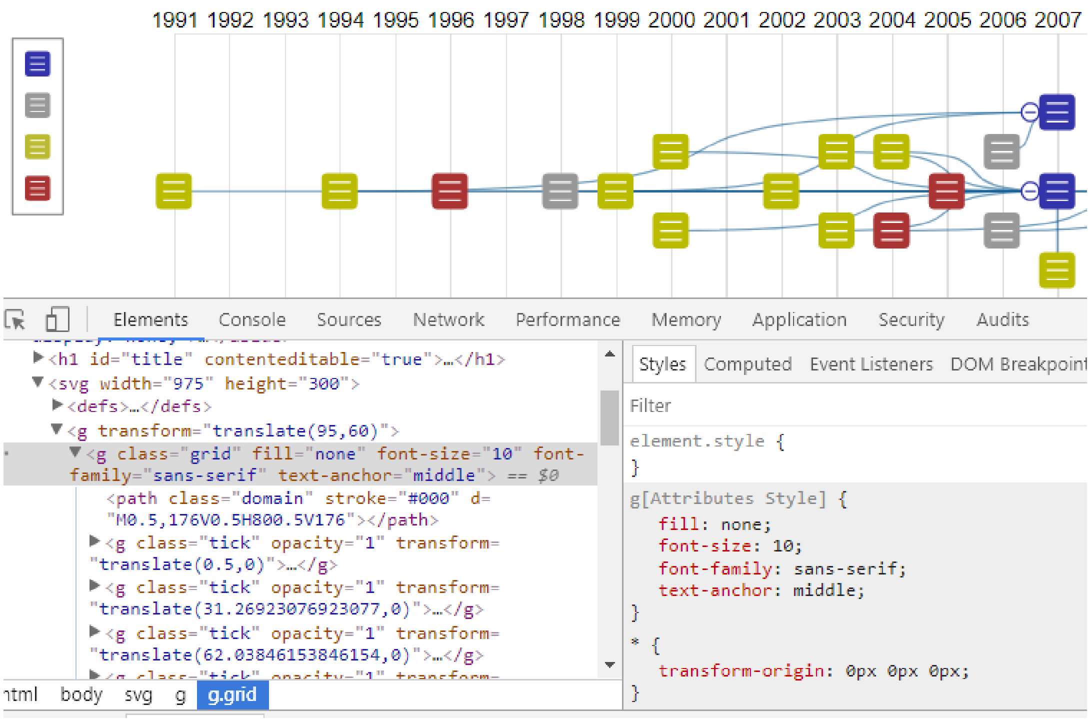

Figure 1.5: The DOM inspector with the style editor integrated in the Chrome browser.

1.2 Introducing zz-structures

Zz-structures were first proposed by Ted Nelson [87, 89] and then revisited in successive works [59, 60] and more recently in [55, 56]: they introduce an intrinsically non-hierarchical, graph-centric system of conventions for data and computing.

As proved in [83], zz-structures are general data structures, since subsume lists, 2D arrays, trees, polyarchies, and all edge-coloured directed multigraphs. Furthermore, they enable to associate semantic interconnections between vertices; manage, in an holistic view, different contextual dimensions which can be simply highlighted on demand; focusing the attention on a specific item, offer local comprehensive view; and, finally, represent a general conceptual model to, formally and informally, describe our knowledge domain.

A zz-structure can be thought as a space filled with cells, called *zz-cells*, connected into linear sequences. Cells are connected together with links of the same color into linear sequences called *dimensions*. A single series of cells connected in the same dimension is called *rank*: a rank is in a particular dimension and a dimension may contain many different ranks. The starting and the ending cells of a rank are called *headcell* and *tailcell*, respectively, and the versus from the starting (ending) to the ending (starting) cell is called *posward* (*negward*).

Preliminary definitions

Preliminary, we remind some basic concepts of graph theory, that we will use below: multigraph, edge-colored multigraph, and degree of a vertex.

Definition 1.2.1. Multigraph - A *multigraph* is a graph where are possible parallel edges, that is, edges that have the same end vertices.

Definition 1.2.2. Edge-colored multigraph - An *edge-colored multigraph* is a triple $ECMG = (MG, C, c)$ where: $MG = (V, E, f)$ is a multigraph composed of a set of *vertices* V, a set of *edges* E and a function $f : E \rightarrow \{\{u, v\} \mid u, v \in V, u \neq v\}$. C is a set of colors, and $c : E \rightarrow C$ is an assignment of colors to edges of the multigraph, where no parallel (i.e. joining the same pair of vertices) edges have the same color.

We note that in an edge-colored multigraph the colors may be substituted by labels; for this reason, a zz-structure may be equivalently defined as a *labelled multigraph.*

Definition 1.2.3. Degree of a vertex - The degree of a vertex x of a graph, denoted $deg(x)$ (respectively, $deg_k(x)$), is the number of edges incident to x, (respectively, of color c_k).

Formal definition of zz-structures and their primitives

There are different ways to formally define a zz-structure. We use the definition proposed in [58], where zz-structure is described as a colored multigraph generated by the union of subgraphs containing edges of a unique color.

Definition 1.2.4. Zz-structure - Consider a set of colors $C = \{c_1, c_2, ..., c_{|C|}\}$ and a family of indirect edge-colored graphs $D = \{d.1, d.2, ..., d.|C|\}$, where $d.k = (V, E^k, f, \{c_k\}, c)$, with $k = 1, ..., |C|$, is a graph such that:

1. $E^k \neq \emptyset$ is the set of edges of colors $\{c_k\}$;
2. $\forall x \in V$, $deg_k(x) = 0, 1, 2$.

Then, $S = \bigcup_{k=1}^{|C|} d.k$ is a zz-structure.

In a zz-structure, V is the set of vertices; $E = \{E^1, E^2, ..., E^{|C|}\}$ is the set of edges; D the set of dimensions. Each vertex of a zz-structure is called *zz-cell* and each edge a *zz-link*. The set of isolated vertices is denoted $V_0 = \{x \in V \ : deg(x) = 0\}$.

A simple example of zz-structure is proposed in Figure 1.6.

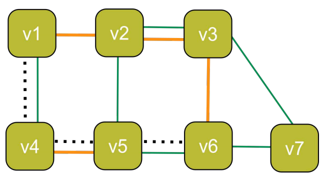

Figure 1.6: An example of zz-structure.

Normal, dotted and thick edges represent respectively green, black and orange colors. Different colors link zz-cells into different spatial dimensions [83], and describe different semantic ties [59].

Definition 1.2.5. Dimension - Given a zz-structure $S = \bigcup_{k=1}^{|C|} d.k$, then each subgraph $d.k$, $k = 1 \ \ldots,\ |C|$, is a distinct *dimension* of S.

Each series of cells sequentially connected in any dimension identifies a *rank* [89].

Definition 1.2.6. Rank - Given a zz-structure $S = \cup_{k=1}^{|C|} d.k$, $\forall$ dimension $d.k = (V,$ $E^k, f, \{c_k\}, c)$, $k = 1, \ldots, |C|$, is called *rank* each of the l_k ($l_k \geq 1$) connected components of $d.k$.

Thus a rank is an indirect graph $R_i^k = (V_i^k, E_i^k, f, \{c_k\}, c)$, where $i = 1, 2, \ldots, l_k$, such that:

1. $E_i^k \subseteq E^k$ and $E_i^k \neq \emptyset$;
2. $deg_k(x) = 1, 2, \forall x \in V_i^k$, where $V_i^k \subseteq V$.

Definition 1.2.7. Ringrank - A *ringrank* is a rank R_i^k, where $\forall x \in V_i^k,\ deg_k(x) = 2$.

Figure 1.7 shows the 3 dimensions and the 5 ranks contained in the zz-structure of Figure 1.6.

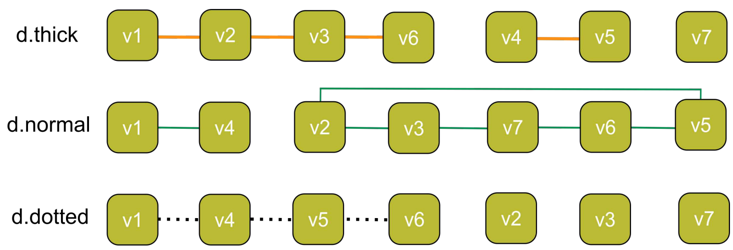

Figure 1.7: The three dimensions and related ranks and isolated zz-cells.

As we may see, each dimension is composed by a set of connected components and a set (eventually empty) of isolated vertices. In our example, *d.thick* is composed of two ranks $\{v_1, v_2, v_3, v_6\}$ and $\{v_4, v_5\}$, and one isolated vertex v_7; *d.normal* is composed of two ranks and no isolated vertex: the first rank is a path $\{v_1, v_4\}$, the second one is a ringrank $\{v_2, v_3, v_7, v_6, v_5\}$; finally, *d.dotted* contains a rank $\{v_1, v_4, v_5, v_6\}$ and three isolated zz-cells v_2, v_3, v_7.

Cells

There exist different typologies of cells:

- *referential* represent the package of different cells;
- *composite* contain more than one type of data;
- *maincell* stands for the whole - denominating one cell as the cell to refer to when we wish to refer to the whole unit. The maincell may be the headcell or tailcell of its rank. A maincell is expected to be connected directly to its supporting cells.
- *compound* contain cells or, in general, zz-structures;
- *positional* do not have a content and thus have a positional or topographical function.

In the following, a **compound** cell is addressed by the following notation [56]: a ring over the letter $\mathring{v}$ denotes a compound cell, and $\mathring{V}$ a set of compound cells. In Figure 1.8 is shown an example containing two typologies of compound cells $\mathring{v}_1$ and $\mathring{v}_2$: the first is composed by a zz-structure and is linked to the cell v_8 by the dimension *d.dotted*; the second groups a set of cells semantically joint from the dimension *d.dashed*, the same that connects v_8 to it.

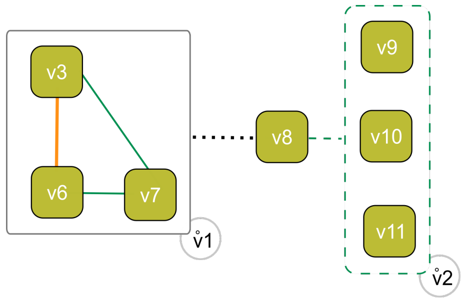

Figure 1.8: Two compound cells are connected with v_8.

Dimensions

Dimensions may be:

- *passive and nominal* receiving and presenting data; linking them through semantic connections.

- *operational*: other dimensions may be operational, programmed to monitor changing zz-structures and events, and calculate and present results automatically. For example, *d.clone* [89] is an operative dimension which connects all clones of a given cell as a single rank. The headcell holds the contents; each clone, when displayed, shows the contents of the headcell. Clones represent an implementation of the transclusion concept at the cell level. This means that the same cell contents can have multiple references at the same time.

Local and global orientation

For any dimension, a cell can have only one positive and only one negative side [89]; of consequence, only one connection in the posward direction, and only one in the negward direction. This ensures that all paths are non-branching, and thus embodies the simplest possible mechanism for traversing links. This idea is formalised in [58] introducing the concept of local and orientation: a vertex has local orientation on a rank if each of its (1 or 2) incident edges has assigned a distinct label (1 or -1).

Definition 1.2.8. Local orientation - Consider a rank $R_i^k = (V_i^k, E_i^k, f, \{c_k\}, c)$ of a zz-structure $S = \cup_{k=1}^{|C|} d.k$. Then, $\exists$ a function $g_x^i : E_i^k \to \{-1, 1\}$, such that, $\forall x \in V_i^k$, if $\exists y, z \in V_i^k : \{x, y\}, \{x, z\} \in E_i^k$, then $g_x^i(\{x, y\}) \neq g_x^i(\{x, z\})$. Thus, we say that each vertex $x \in V_i^k$ has a *local orientation* in R_i^k.

Thus, local orientation is a property related to each vertex of a rank. The vertices of the zz-structure also have a global orientation, i.e., we can extend the previous property to all the ranks and dimensions. Moreover, all the local choices of orientation are consistent.

Definition 1.2.9. Global orientation - Assume a zz-structure $S = \bigcup_{k=1}^{|C|} d.k$ has $l = \sum_{k=1}^{|C|} l_k$ ranks $R_i^k = (V_i, E_i^k, f, \{c_k\}, c)$, $i = 1, \ldots, l_k$, and $k = 1, \ldots, |C|$. Then, S has *global orientation* iff, $\forall \{x, y\} \in E_i^k$, $\forall i = 1, \ldots, l_k$, and $\forall k = 1, \ldots, |C|$, we have $g_x^i(\{x, y\}) \neq g_y^i(\{x, y\})$.

Definition 1.2.10. Posward and negward directions - Given an edge $\{a, b\} \in E_i^k$, we say that $\{a, b\}$ is in *posward* direction from a in R_i^k, and that b is its *posward cell* iff $g_a^i(\{a, b\}) = 1$; else $\{a, b\}$ is in *negward* direction and a is its *negward cell*.

If we focus on a vertex x, $R_i^k = \ldots x^{-2}x^{-1}\, xx^{+1}x^{+2}\ldots$ is expressed in terms of negward and posward cells of x: x^{-1} is the negward cell of x and x^{+1} the posward cell. We also assume $x^0 = x$. In general x^{-i} (x^{+i}) is a cell at distance i in the negward (posward) direction.

Definition 1.2.11. Headcell and tailcell - Given a rank $R_i^k = (V_i^k,\ E_i^k, f, \{c_k\}, c)$, a cell x is the *headcell* of R_i^k iff $\exists$ its posward cell x^{+1} and $\nexists$ its negward cell x^{-1}. Analogously, a cell x is the *tailcell* of R_i^k iff $\exists$ its negward cell x^{-1} and $\nexists$ its posward cell x^{+1}.

1.3 Zz-views

Zz-structure generates a pseudo-space that is somewhat comprehensible visually; there is no canonical viewing mechanism for zz-structures [89]. In this section, we introduce some existing relevant zz-views proposed in literature with the intention of extending these proposals with new and original zz-views to enhance the exploration of the information domains of the two case studies treated in Chapters 2 and 3.

1.3.1 H and I views

Typical views are the two 2D cursor-centric views called the H-view (or column view) and I-view (or row view).

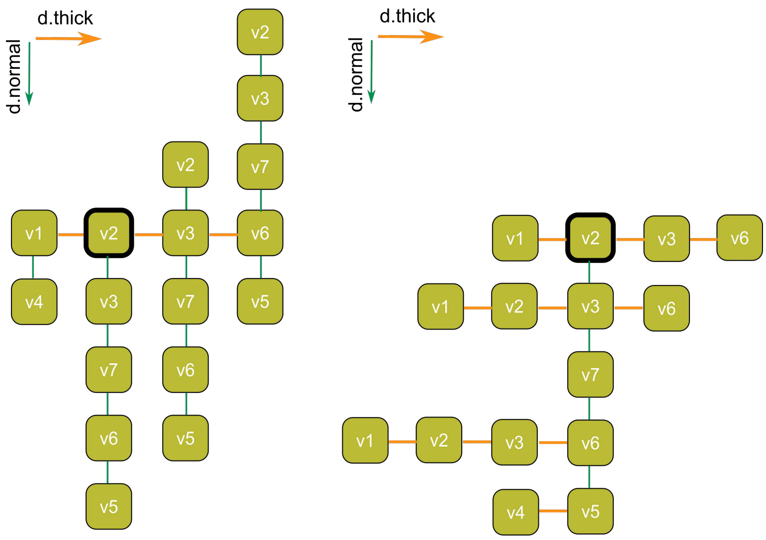

Figure 1.9: The H-view (left) and the I-view (right) centered on the vertex *v2* of the zz-structure of Figure 1.6 relative to the *d.thick* (horizontal) and *d.normal* (vertical) dimensions. In this example the ranks along the *d.normal* dimension are not treated as ringranks but as simple ranks.

These two views make use of 2 spatial dimensions at a time, and locally flatten a subset of the neighborhood around a cursor (i.e. a selected node). More specifically they visualize the subset of nodes that are connected to the cursor's node via edges along the two chosen dimensions. This subset of nodes is embedded in a (non-Euclidean) 2D manifold, which is then displayed in a flat, 2D view [83] as visible in the example of Figure 1.9. In this Figure we can note how multiple instances of a vertex appear as it can be reached along the chosen dimensions starting from different vertices. H and I-views generally enable users to move the cursor in the four directions (up/down and left/right) and change the horizontal or vertical dimension in order to navigate through all available ranks connected directly/indirectly to the selected vertex.

Since H-views and I-views do not, in general, make use of all the space available in a 2D view, McGuffin in [82] proposed two corresponding augmented views, called respectively H^+-view and I^+-view, in order to possibly fill the available empty slots with cells reachable along one of two selected dimensions. In particular, a H^+-view (respectively I^+-view) is built, starting from the H-view (respectively I-view), by possibly filling each empty slot with a cell connected to the adjacent ones along the horizontal (respectively vertical) dimension. The process is then repeated for the other dimension in order to further fill the view. Figure 1.10 shows the H^+ and I^+-view obtained by enriching the standard H and I-views of Figure 1.9.

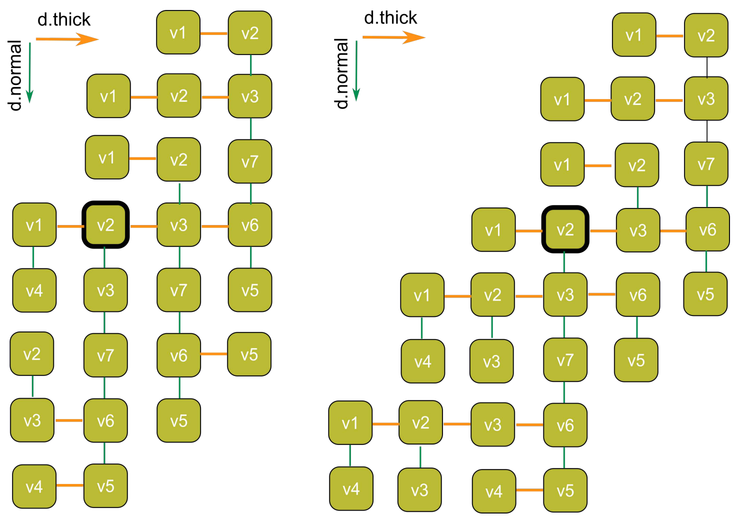

Figure 1.10: The augmented H^+-view (left) and the I^+-view (right) centered on the selected vertex *v2* of the zz-structure of Figure 1.6 relative to the *d.thick* (horizontal) and *d.normal* (vertical) dimensions. In this example the ranks along the *d.normal* dimension are not treated as ringranks but as simple ranks.

These views have been introduced in [83, 82], and formally defined and extended into n-dimensions H-views and I-views in [57, 59]. Figure 1.11 shows an example of 3-dimensions extended H and I-view which includes all the dimensions of the zz-structure of Figure 1.6. All the vertices accessible along the third dimension (*d.dotted*) starting from the cells belonging to the main rank (horizontal in the H-view and vertical in I-view), are made visible and connected along the diagonal direction.

Drawbacks of H and I-views and their variants are represented by a possible complex representation of the zz-structure due both to the presence of multiple instances of a same vertex, and to the significant reorganization of the view after the movement of the cursor or the change of one of the selected dimensions.

For these reasons, in literature other views have been proposed and specific zz-views are introduced to deal with particular information domains, as those presented in the case studies in Chapters 2 and 3.

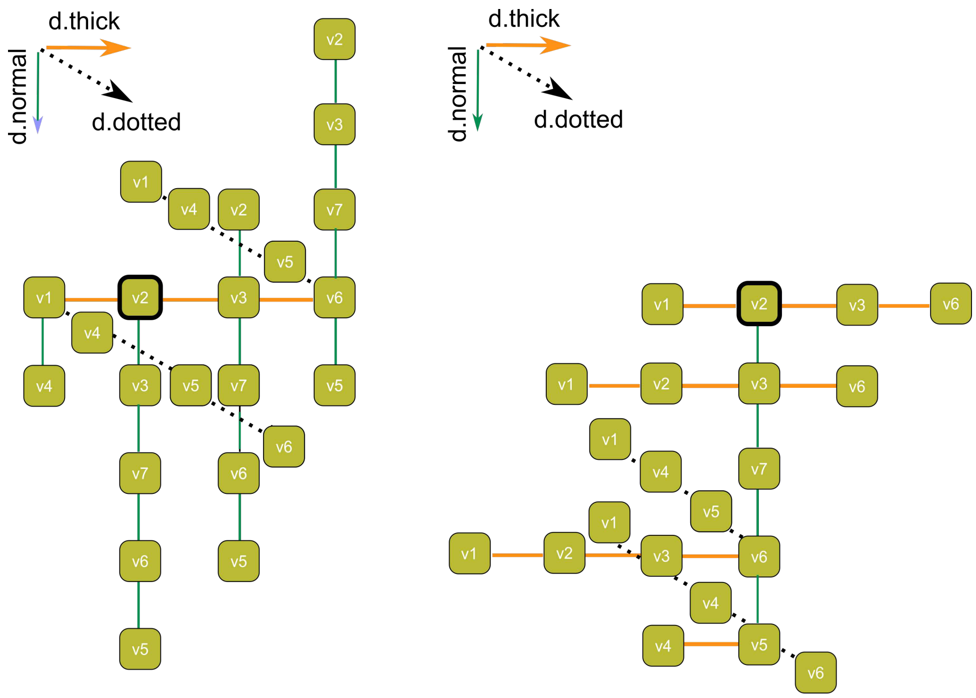

Figure 1.11: The 3-dimensions H-view (left) and the 3-dimensions I-view (right) centered on the vertex *v2* of the zz-structure of Figure 1.6 relative to all its dimensions. In this example the ranks along the *d.normal* dimension are not treated as ringranks but as simple ranks.

1.3.2 Star-views and the *m*-extended star-views

A further two views are formally introduced in [59]: the star-view and the *m*-extended star-view. A star-view focused on a vertex v enables user to directly explore, in posward direction, the adjacent vertices, along all dimensions.

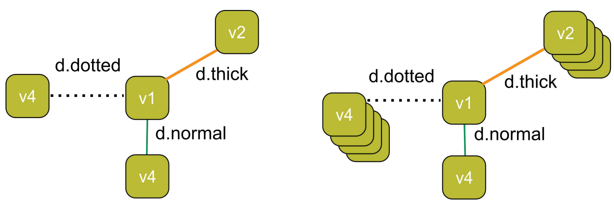

Figure 1.12: A star-view focused on vertex *v1* (left) and a *4*-extended star-view focused on the same vertex *v1*. The *4* extension is reached only by *d.dotted* and *d.thick* dimensions.

The *m*-extended star-view presents, for all dimensions, up to m vertices in posward direction, along all dimensions. Figure 1.12 shows these views for the considered zz-structure, centered on vertex *v1*.

These types of views give users an overall visualization on how the selected vertex is connected along all dimensions and permit to move the cursor posward to any visible vertex but do not allow negward direction movements unless a mechanism is provided to "invert" the view and display the vertices, along all the dimensions, which have links to the selected node.

1.3.3 List-views

A simple new zz-view, useful to navigate along a single rank of a zz-structure, is the list-view. It displays an ordered list of the connected cells. This typology of view has been used in our case studies, offering users mechanisms to change dimension/rank and to re-order the list according to different criteria. Figure 1.13-right shows an example of list-view applied to a rank of the *d.normal* dimension of the zz-structure of Figure 1.6, while Figure 1.13-left shows simple navigation mechanisms.

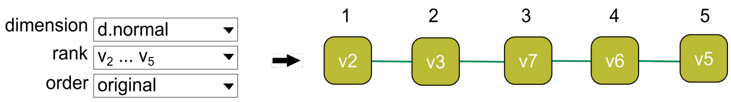

Figure 1.13: A list-view relative to the 2nd rank of the *d.normal* dimension of the zz-structure of Figure 1.6. In the example the selectors for the dimension, rank and order have been included.

1.3.4 Deep-views

Another new proposal of zz-view is the deep-view. A deep-view can be applied to a single rank or to compound cells. In the following we introduce the definitions of the deep-view in the two cases.

Definition 1.3.1. Deep view - Given a zz-structure S, a rank of cells $\{v_1, \ldots, v_m\}$, joint along the dimension $d.k$ and with maincell v_1 (respectively v_m), then the *deep view* of *focus* v_1 (respectively v_m) and dimension $d.k$ displays a graph, where:

- $V = \{v_1, \ldots, v_m\}$;
- $E = \{(v_1, v_j)|j = 2, \ldots, m\}$ (or resp. $E = \{(v_j, v_m)|j = 1, \ldots, m-1\}$).

In other words, the deep-view of a rank displays a link from its maincell (headcell or tailcell) to all other cells of the rank, enabling users to directly explore each of them moving from the focused cell. Figure 1.14 (left) shows two examples of deep-views of a given rank (top-left), respectively centered on the headcell v_1 and on tailcell v_6.

The deep-view can also be applied to compound cells:

Definition 1.3.2. Deep view for compound cells - Given a zz-structure S, and:

1. a compound cell $\mathring{v}$, constituted by a set of cells $\{v_1, \ldots, v_m\}$, joint along the dimension $d.k$;
2. a cell $v \in V$ and an edge in the same dimension $d.k$, which links $(\mathring{v}, v)$ (or indifferently, $(v, \mathring{v})$);

then the *deep view* of *focus* v and dimension $d.k$ displays a graph, where:

- $V = \{v, v_1, \ldots, v_m\}$;
- $E = \{(v_j, v)|j = 1, \ldots, m\}$ (or resp. $E = \{(v, v_j)|j = 1, \ldots, m\}$).

Figure 1.14 (right) shows a deep-view of focus v and dimension $d.k$ applied to the compound cell $\mathring{v}$. A specific application of this view will be presented in the end of this chapter to offer a visual representation of citing relations between papers of the bibliography about the zz-structure. This model will be extended and generalized in Section 2.5 in the context of the VisualBib platform.

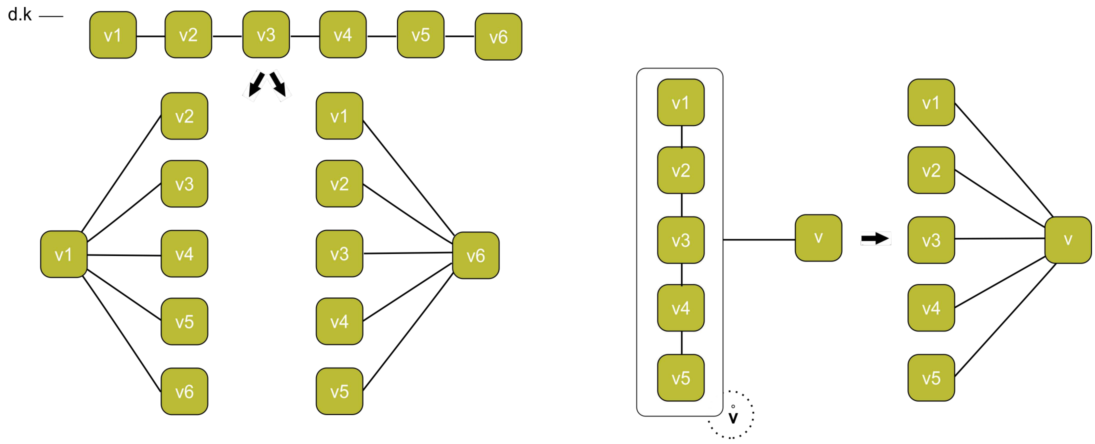

Figure 1.14: On the left two examples of deep-views of a rank along dimension $d.k$ (top-left), respectively centered on the headcell v_1 and tailcell v_6. On the right a deep-view of focus v and dimension $d.k$ applied to the compound cell $\mathring{v}$.

1.3.5 Narrative-views

Another new proposal is the narrative-view; it is applicable to zz-structures in the case that each element can be attributed to one or more agent(s) and has a time mark associated to them. An application of this view will be presented at the end of this chapter where an interactive narrative-view representation of the scientific production about the zz-structures will be illustrated. A generalized model of this view has been also integrated in the VisualBib platform presented in Chapter 2.

Definition 1.3.3. Narrative view - Given a zz-structure S consisting of:

- a set $V = P \cup A \cup T$ of elements $p_1, p_2, \dots, p_{|P|} \in P$, $|P| \neq 0$, each of them attributable to specific agent(s) $a_1, a_2, \dots, a_{|A|} \in A$, $|A| \neq 0$, and associated to a specific time mark $t_1, t_2, \dots, t_{|T|} \in T$, $|T| \neq 0$;
- a set of dimensions $\{d.time, d.a_1, \dots, d.a_{|A|}\}$ where:
  - $d.time$ is the time dimension containing a set of parallel ranks $d.time = \bigcup_{i=1}^{|T|} R_i^{t_i}$ each linking the elements in P associated to the time mark t_i;
  - $d.a_i$ are the agent(s) dimension(s), each of them linking the elements in P attributed to agent a_i;

a *narrative view* displays:

1. an horizontal timeline $t_1, t_2, \dots, t_{|T|}$, drawn left to right; under each $t_i, i = 1, 2, \dots, |T|$, in vertical, is positioned the parallel rank $R_i^{t_i}$, belonging to the time dimension $d.time$, and containing all the items $p \in P$, marked by t_i;
2. $\forall p \in P$ the dimensions $d.a_1, d.a_2, \dots, d.a_{|A|}$ as linear paths that links each agent to all its elements. The linear paths are time ordered;

Figure 1.15 shows an example of narrative view (right) of a small zz-structure (left) composed by 13 vertices (elements $p_1, \dots, p_7$, agents a_1, a_2, a_3 and time marks t_1, t_2, t_3) and 4 dimensions ($d.time$, $d.a_1, d.a_2, d.a_3$). The narrative view is built according to the following guidelines:

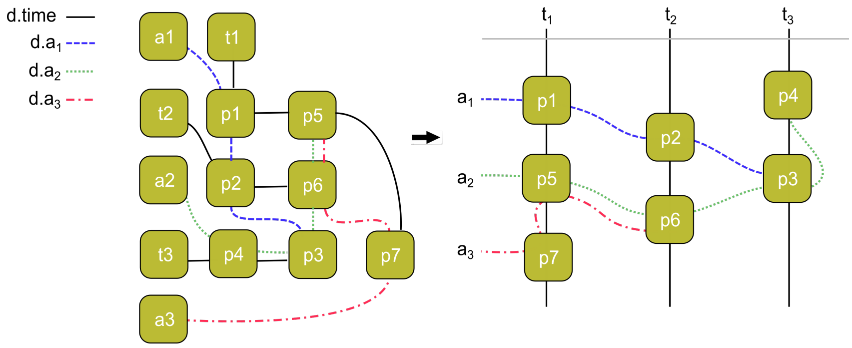

Figure 1.15: An example of narrative view (right) of a zz-structure (left) of 13 vertices and 4 dimensions. In the narrative view the t_i vertices are rendered as labels placed along the time axis, *d.time* dimension's paths are replaced by a vertical grid, the agents' vertices are displayed as labels on the left of the view and agents' paths connect elements in time order.

- the t_i vertices are displayed as labels placed along the horizontal time axis; in this example the time marks are equidistant but this is not a requirement of the view;
- the links of the *d.time* dimension are not explicitly represented being replaced by a vertical grid, representing the ranks $R_i^{t_i}$;
- the agents' vertices a_i are displayed as labels on the left side of the view; in general, if the first element of a certain agent is associated to a time mark $t_i, i > 1$, the corresponding agent's label is placed in the column t_{i-1};
- the ranks relative to all the dimensions $d.a_i$ are ordered by time and the links are stylized in a common way: in order to distinguish the path of a specific agent, user interaction mechanisms can be provided to highlight the the involved elements and links (e.g. by coloring them when user clicks on the label of an agent);
- the method of positioning the elements along the vertical axis should avoid the overlapping of labels and elements.

1.3.6 Bubble-views

These new zz-views have been introduced to model the holistic representation of the multimedia catalog AppInventory illustrated in Chapter 3.

A bubble-view is focused on the maincell (headcell or tailcell) of a rank and displays it as a set (represented by a circular shape or other closed curve) which contains all the remaining cells of the rank.

Definition 1.3.4. Bubble view - Given a zz-structure S, a rank of cells $\{v_1, \ldots, v_m\}$, joint along the dimension $d.k$ and with maincell v_1 (respectively v_m), then the *bubble view* of *focus* v_1 (respectively v_m) and dimension $d.k$ displays a circle, marked by label v_1 (resp v_m), containing the set of cells $B = \{v_2, \ldots, v_m\}$ (or resp. $B = \{v_1, \ldots, v_{m-1}\}$).

This view visually represents the relation between the cells of a rank along the dimension $d.k$, as the belonging to a common set represented by the headcell of the rank. Figure 1.16 shows two examples of bubble-views of a rank (top), respectively centered on the headcell v_1 (bottom-left) and on tailcell v_6 (bottom-right).

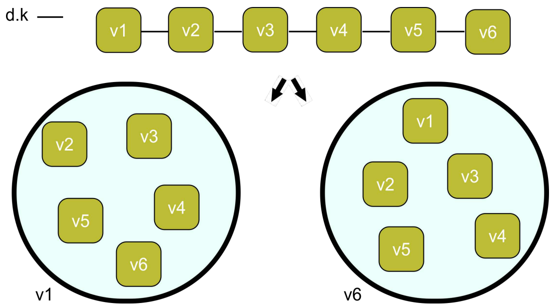

Figure 1.16: Two examples of bubble-views of a rank along dimension $d.k$ (top), respectively centered on the headcell v_1 (bottom-left) and tailcell v_6 (bottom-right).

The bubble-views enable hierarchic representations of zz-structures along specific dimensions as in the example of Figure 1.17 where a 2-dimension zz-structure is represented by means of two levels of nested sets. The size of each circle is set to be proportional to the number of elements contained in it.

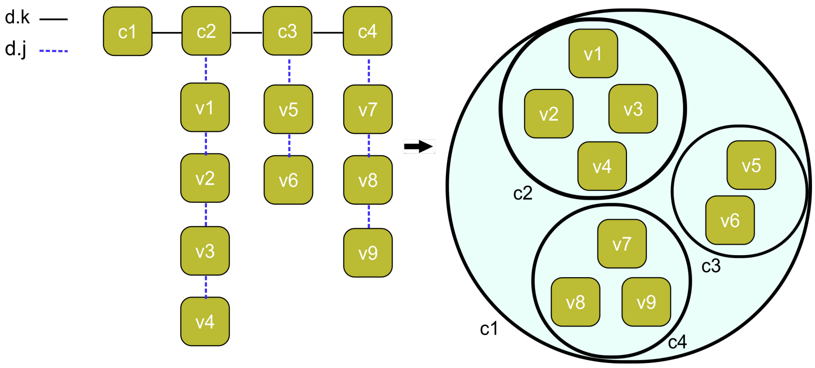

Figure 1.17: A two-level nested bubble-view of the zz-structure along the dimensions $d.k$ and $d.j$.

This two-level nested bubble-view model has been applied and generalized in the context of the AppInventory project presented in Chapter 3. In that context, since multiple dimensions are considered, any cell in common to different ranks is duplicated in the view in order to avoid intersections between inner sets, as illustrated in Figure 1.18.

1.4 Building a bibliography on zz-structures

In order to document the advances in the research on zz-structures we investigated related work in the last 20 years and collected the significant scientific papers, at best of our knowledge, about the topic, including case studies and applications in different domains. Up to now, we have selected a set of 60 papers, gathering the full-texts, extended metadata (date of publication, paper type, category, author names and their affiliations, abstract, keywords, publisher, ...) and cross-citation metadata. All materials have been organized in a database in order to make them available in a dedicated website `http://zzstructure.uniud.it`, enabling users to consult and order the list of papers with different criteria. Looking for an alternative representation of the bibliography we considered the zz-structure itself as the base for a semantic organization of the materials.

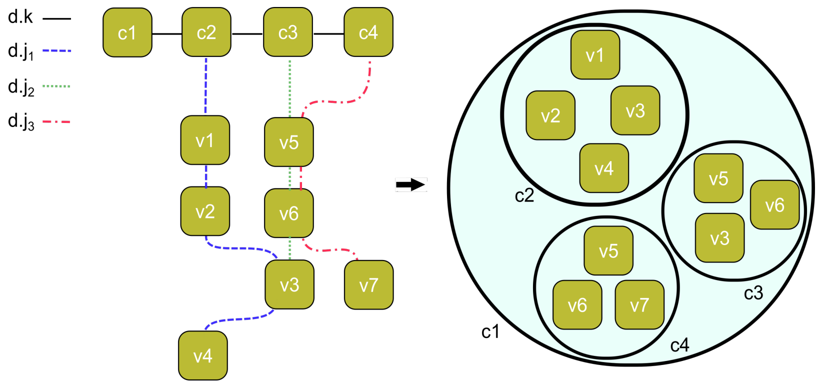

Figure 1.18: A two-level nested bubble-view of the zz-structure along the dimensions $d.k$ and $d.j_1$, $d.j_2$, $d.j_3$. Cells in common to the considered ranks (v_3, v_5, v_6 in this example) appear duplicated in the view in order to avoid overlaps between bubbles.

The idea was to offer an interactive view of the overall bibliography that highlights the temporal order of the papers, the authors involved and their collaborations over the time and the citation relationships between papers, allowing, at the same time, an easy access to their metadata. The development of this first prototype has subsequently inspired the idea for VisualBib, a platform for dynamically building, refining and sharing custom bibliographies, we will present in next Chapter 2 and formalize in Section 2.4.

1.4.1 A zz-structure model for the zz-structure's bibliography

We present here the zz-structure used to model the complete bibliography on zz-structures.

In brief, the zz-structure is composed by:

- a set of vertices $V = \{P, A, Y, DE, FT, BI\}$ where:
  - P is a set of 60 bibliographic references (papers or websites);
  - A is a set of 51 authors;
  - Y is a set of 22 time marks representing each year between 1998 and 2019;
  - DE is a set of 60 composite cells containing details related to papers, such as title, authors, etc., and links to external resources (DOI, Scopus, ...);
  - FT is a set of 60 full texts of the papers;
  - BI: the set of 60 BibTex documents associated to the references;
- a set of dimensions $D = \{d.a_1, \ldots, d.a_{51}, d.time, d.citedby_1, \ldots, d.citedby_{60}, d.citing_1, \ldots, d.citing_{60}, d.details, d.full\}$ where:
  - $d.a_1, \ldots, d.a_{51}$ identify the dimensions associated respectively to each author $a_1, \ldots, a_{51} \in A$. Each dimension connects the papers of the corresponding author ordered by publication time. The maincell of each dimension $d.a_i$ is the vertex a_i;
  - $d.time$ connects the papers having in common the publication year and is therefore constituted by the 22 ranks $d.time_{1998}, \ldots, d.time_{2019}$. The maincell of each rank $d.time_y$ is the vertex $y \in Y$;

- $d.citedby_1, \ldots, d.citedby_{60}$ dimensions connect all the papers cited by paper $p_1, \ldots, p_{60} \in P$ respectively, which is also the tailcell of the related dimension;
- $d.citing_1, \ldots, d.citing_{60}$ dimensions connect the papers which cite the paper $p_1, \ldots, p_{60} \in P$ respectively, which is also the headcell of the related dimension;
- *d.details* dimension connects each paper with the $de \in DE$ composite cell containing some metadata of the paper; *d.details* dimension is therefore partitioned in 60 parallel ranks, one for each paper;
- $d.full$ dimension connects each paper with a full text $ft \in FT$ cell containing the full text version of the paper itself; $d.full$ dimension is therefore partitioned in 60 parallel ranks, one for each paper;
- *d.bibtex* dimension connects each paper with a text $ft \in BI$ cell containing the BibTex document associated to the paper; *d.bibtex* dimension is therefore partitioned in 60 parallel ranks, one for each paper.

1.4.2 Zz-views for the zz-structure's bibliography

In order to visualize the bibliography we implemented a specialized version of the narrative view, formally defined in Subsection 1.3.5, which initially displays on a plane:

1. the set P of the 60 references (papers or websites);
2. the set A of the 51 authors' names, corresponding to the headcells of the dimensions $d.a_1, \ldots, d.a_{51}$;
3. the set of 22 numeric labels in Y, representing the years in the interval [1998,2019], disposed along the horizontal axis and marked with vertical grid lines;
4. the links relative to the dimensions $d.a_1, \ldots, d.a_{51}$ connecting the relative papers.

Figure 1.19 shows the initial visualization of the narrative view. In addition to the main narrative-view a series of deep-views are dynamically generated to show the citation relationships between the papers. In particular a couple of deep-views, associated respectively to the *d.citedby* and *d.citing* dimensions, highlight the cited papers and the citing papers (see Figure 1.20). The deep-views are normally hidden to improve the readability of the narrative-view but they can be made visible on demand, as described in the following. The narrative-view was made interactive through the management of a series of events. Users can interact with it in the following ways:

1. moving the cursor over an author's label a_i: all the papers in the dimensions $d.a_i$ and the relative links are colored in red in order to highlight the production of the author;
2. dragging vertically the icons of a paper or an author's label in order to change the layout of the narrative view; the new configuration is maintained for the current session only but can be saved by the administrator to offer all user an improved representation of the narrative view;
3. clicking on a paper p_i in order to generate and visualize two specific deep-views (see Section 1.3.4): the first related to the dimension $d.citedby_i$ and focused on the tailcell p_i; the second related to the dimension $d.citing_i$ and focused on the headcell

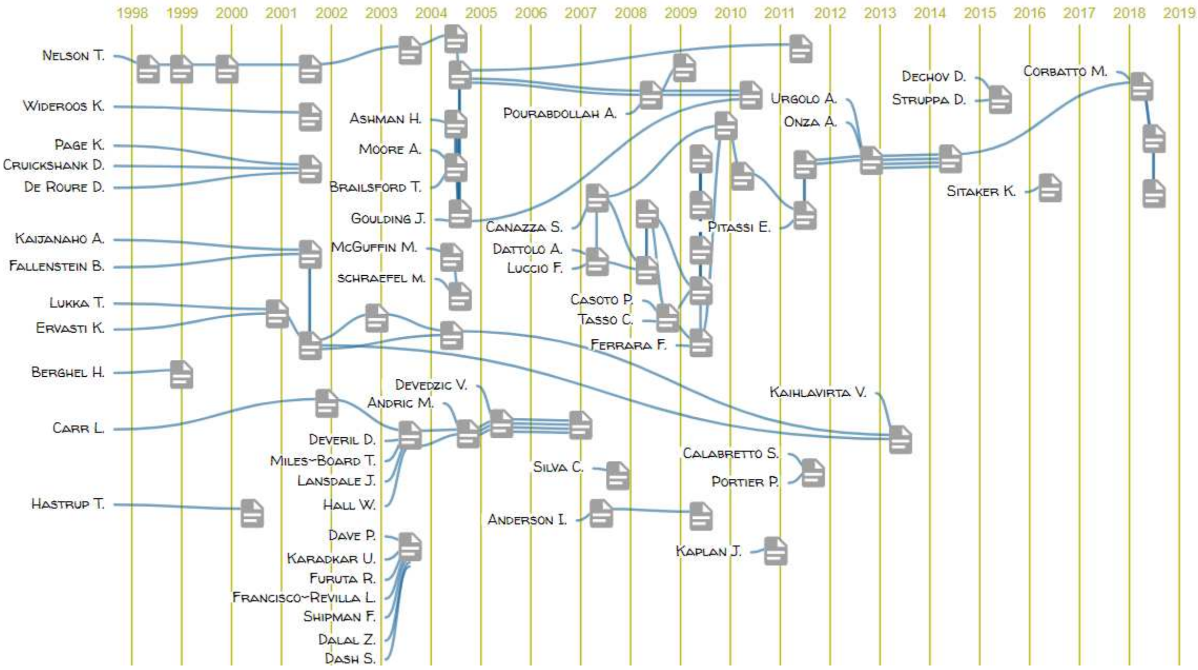

Figure 1.19: The initial picture of the narrative view showing all the papers in the bibliography, the names of the authors, the labels of the years together with the corresponding grid and the links relative to authors' dimensions.

p_i. These deep-views are visible in Figure 1.20 (left) respectively colored in green and orange;

4. double clicking on a paper p_i: a view of the composite cell de_i becomes visible. It displays the list of the authors, the publication year, title of the paper, some bibliographic metadata (journal name or book title, volume, pages, publisher, ISBN, ISSN, ...) and the references to external resources (DOI, Scopus and WOI pages) together with links to explore the related ft_i (full-text) and bi_i (BibTex) cells.

1.4.3 Generating the narrative view

In this section we introduce the guidelines for the generating the narrative view for the bibliography on zz-structures. These specifications starts from those introduced in Subsection 1.3.5 and will be generalized in Section 2.5.3, in the context of the VisualBib platform. Without entering in details of the algorithms adopted, we describe a series of requirements taken in account:

- the proper arrangement of the papers' icons on the 2D plane in order to satisfy some visual cues:
  - a correct positioning along the horizontal temporal axis; this has been achieved fixing a linear scale to map the date of publication (year and month) to a specific x position;
  - the prevention of icons' overlaps: a minimum vertical distance has been established to assure appropriate spacing in the positioning of papers with nearby publication years;

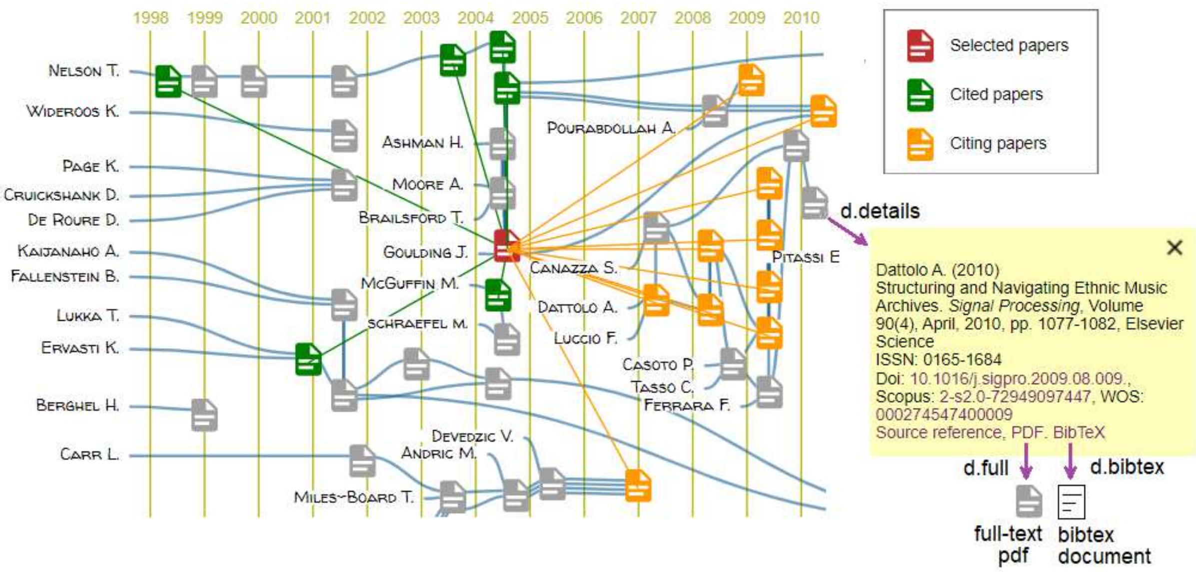

Figure 1.20: The new zz-views during interactions with the narrative view: on the left the two deep-views related respectively to the $d.citedby_i$ (green) and $d.citing_i$ (orange) dimensions focused on the clicked cell p_i. On the right-bottom the view of the composite cell de_i containing some metadata of the paper, the references to external resources and links to access the related ft_i and bi_i cells.

  - the minimization of the distance along the vertical axis of the papers' icons having in common at least one of the $d.a_1, \ldots, d.a_{60}$ dimensions in order to avoid large fluctuations in the connection paths;
  - the avoidance of overlapping papers' icons p_i with paths of extraneous authors' dimensions $d.a_j : p_i \notin d.a_j, j = 1, \ldots, 51$.

- the arrangements of the authors' labels: the criterion adopted, previously introduced in Subsection 1.3.5, was to position the headcell on the column relative to the year before the publication date of the first paper of the author. For an aesthetic reason, all the labels relative to authors with first publication year in the range [1998,2001] are positioned in correspondence of year 1997. The vertical position of each label is set to avoid overlaps with papers' icons and other authors' labels;

- the drawing of the lines relating to the $d.a_1, \ldots, d.a_{51}$ dimensions: *smooth cubic Bezier curves* have been adopted to connect consecutive cells. The curve parameters have been calculated in order to possibly avoid overlapping paths for different authors.

Despite these guidelines, the resulting automatic positioning of papers and authors was not completely satisfactory, so we have provided a mechanism to move the icons and labels vertically to allow the user to freely modify the layout of the diagram. Although the arrangement is not maintained between work sessions, it can be made permanent by the administrator in order to propose users an effective initial layout. Figure 1.19 shows the result of a manual adjustment of the automatic disposition of items described above. VisualBib, as described in the relative Chapter 2, adopts a completely automatic positioning algorithm releasing some constraints (i.e. not overlaps of paths) and introducing new ones, like symmetric positioning of papers.

1.5 Conclusions

In this chapter we presented some concepts, models, guidelines and formalizations concerning information visualization and zz-structures. The idea is to apply principles and techniques of information visualization to zz-structures through the mechanism of zz-views. In particular Section 1.1 gave a general introduction to visual methods for information visualizations and on technologies for their implementations in the context of Web applications.

The Sections 1.2 and 1.3 introduced the zz-structures and new proposals of zz-views to be applied to the considered case studies.

In order to start investigating this approach we developed a prototype of Web application with the aim to offer an holistic and interactive view of a "static" bibliography about the zz-structure literature. Section 1.4 described this project starting with a model of the considered bibliography based on the zz-structure and two zz-views based on the previously introduced deep-view and narrative-view.

Both the new proposals of zz-view introduced in the second part of Section 1.3 and the interactive bibliography about the zz-structure literature described in Section 1.4 represent original contributions of this thesis.

The next chapter will further investigate this approach through the modeling and implementation of a generalized version of this application in order to represent, manage and analyze custom scientific bibliographies with real-time data retrieval from external bibliographic indexes.

2

VisualBib: building, refining and analyzing scientific bibliographies

The exploration of the scientific literature for creating and saving, in a reusable format, significant scientific references on specific research topics is a common task for researchers. The searches are generally carried out on big citation indexes like Scopus, Web of Science (WOS), CrossRef, Google Scholar, Microsoft Academic, OpenCitations and others, by specifying a set of keywords, the title of a publication or the author names. The results are generally presented in a long list of items, where it is not simple to identify the relations between papers (for example, co-authors, co-citations, temporal order of publications), the typologies of publications (journal, conference, book) or to get a general idea of a specific author's production. Furthermore, each subsequent search brings new results; attempting to aggregate the data would require a considerable effort for the researcher who should manually examine them, find the connections and discard duplicate entries in order to consolidate a set of significant papers.

The idea of adopting visual representations, to show bibliographic data and support users in their analysis and interpretation, has been widely studied [65] and several tools for the visualization of citation networks have been proposed [94]. Unfortunately, all these tools work on bibliographic datasets, which must be retrieved in advance from specific citation indexes. They do not manage multiple sources of data, except through a manual merge of specific datasets; they do not offer rapid methods to share a bibliography.

Our proposal, which represents an original contribution of this thesis, is to offer a new Web application called VisualBib, that interfaces directly with four large bibliographic data providers through API services in order to retrieve updated information about papers, authors and citations.

From a task-oriented point of view, we have identified three main purposes for using VisualBib:

- **The visual representation of bibliographies** through an overall and interactive view that highlights the semantic connections between authors and papers, the distribution of publications and publication types over time and some collaboration metrics. The proposed VisualBib's main interface, called *narrative view*, is presented in Section 2.2 and formally defined in Section 2.5.

- **The support for the creation of bibliographies** through the real-time query of multiple bibliographic indexes, the selection and integration of the retrieved data, the exploration of the citation networks, the importing of external BibTeX archives and the progressive refinement of the bibliography.

- **The sharing of bibliographies** via the Web, their saving on a cloud space, their embedding in external Web pages or their exporting in a standard format.

VisualBib has been designed and implemented as an *personal assistant for researchers* who would like to represent, communicate and share their selections of the scientific production in a certain topic/domain related to a given author (or set of authors); it represents a visual tool able to query in real-time different bibliographic indexes and provide visual analytics insights.

A first prototype of VisualBib has been introduced in [46, 55] together with some initial experimental results. A successive work [56] describes formally and informally the VisualBib app focusing on:

- formal description of VisualBib in terms of zz-structures, with the formal definitions of two new *deep* and *narrative* views;
- integration of the two new bibliographic indexes, CrossRef and Orcid, used in a combined way;
- introduction of new features, like *MatchAuthor*; *Import/Export BibTeX*; the insertion of a *histogram*, representing the distribution over time of the publications; a *stacked area chart*, illustrating the distribution of the publications types; and the possibility of *multiple selection of authors* to highlight the publications in common and their number;
- proposal of some new graphic aspects of the application, like the organization of the search section, the disposition of the items in the narrative view, the highlight of search result in the view and in the list of papers.
- comparative analysis of the API provided by the major current bibliographic indexes;
- a new quantitative and qualitative evaluation, on larger group of participants, and on a more extensive and articulated questionnaire.

A dedicated Website `http://visualbib.uniud.it` documents the evolution of the application and allows interested people to use it.

The rest of the chapter is organized as follows: Section 2.1 discusses related work, emphasizing open issues and challenges in visual representation of bibliographies; Section 2.2 presents our tool, VisualBib, introducing its basic functionalities and user interface; Section 2.3 presents a list of low-level data analysis tasks supported by our tool in comparison with existing bibliographic indexes and analyzes the API services offered by them; Section 2.4 proposes a formal semantic data model of VisualBib based on zz-structures; Section 2.5 documents the application of the two new original zz-views defined in Section 1.3, the *deep-views* and the *narrative-views*, in order to generate a representation of the bibliography and its citation network; Section 2.6 presents a user case scenario to show how VisualBib can support an author in the import, refinement and export of a bibliography starting from a set of references provided in a BibTeX archive; Section 2.7 describes the architecture of VisualBib and some implementation details of the modules; Section 2.8 illustrates a comparative (VisualBib-Scopus) quantitative and qualitative user evaluation of VisualBib; Section 2.9 presents the last version 3.0 of VisualBib which introduces several new features for the analysis of a bibliography and its management; this

Applications	Views	Implementation	Real-time data	Data Integration	Under Development
CiteSpace [42] 2004-today	Interactive visualizations of structural and temporal patterns	Stand-alone	Pre-built, static dataset	No	Active
PaperLens [75] 2004-2005	Views across papers, authors and references	Stand-alone for Windows	Pre-built, static dataset	No	No
BiblioViz [91] 2006	Table and network	Stand-alone	Pre-built, static dataset	No	No
CiteWiz [62] 2007	Author, citation, and metadata views. Concept map for keywords	Stand-alone for Windows	Pre-built, static dataset in XML-based format	No	No
VOSviewer [95] 2007-today	Label, density, cluster density and scatter views	Java stand-alone	Pre-built, static dataset + limited API access	Partially	Active
PaperCube [37] 2009-2010	Views based on graphic, hierarchy, and timeline structures	Web application	Pre-built, static dataset	No	No
Cybis [53] 2011	3D cylinder view	Java Web application	Pre-built, static dataset	No	No
Citeology [81] 2012	Generalized fisheye view	Java applet	Pre-built, static dataset from the ACM DL	No	No
PivotPaths [61] 2012	Interactive pathways	Web application demo	Pre-built, static dataset from MS Academic Search	No	No
CitNetExplorer [93] 2014	Citation networks	Java stand-alone	Pre-built, static dataset generated by WOS	No	No
VisualBib 2017-today	Narrative views	Web application	By API from multiple sources	Yes	Active

Table 2.1: Comparing some technical aspects of the eleven visual tools.

section also illustrates the results of an evaluation study and the analysis of a sample bibliography carried out in the new version. Conclusions end the chapter.

2.1 Related work

A general survey [65] examines 109 different visual approaches to analyze scientific literature and patents, that came to light between 1992 and 2016; this work, together with an interactive visual survey [73] of 400 different techniques for text visualization and with a graphical review [52] of the research on visual languages from 1995 to 2014, highlights the fundamental role of visual representations for a meaningful use of publications' metadata for scientific communities. Starting from these studies and from the rest of current literature, we focus our attention on:

- a set of interesting visual tools, which propose graphic representations of bibliographic data and support researchers in exploring bibliographies. However, it must be said that the purpose of these tools is the visual analysis of large bibliographic dataset in order to cluster information and highlight relations and data patterns, while VisualBib is a tool conceived to support researchers in building up and managing small and medium-sized bibliographies as dynamic spaces where it is possible to add, delete, explore and merge new data. For this reason, the application we propose is not directly comparable with these tools; hence, we discuss and compare only some specific aspects, listed in Tables 2.1 and 2.2;
- ten, widely-used bibliographic indexes, as, for example, AMiner, Google Scholar, MS Academic, Scopus, WOS.

Comparison with ten visual tools Table 2.1 compares ten tools considering the proposed views, the typology of implementation (Web or stand-alone application), the modality of retrieving data (using pre-built datasets or querying bibliographic indexes in real-time) and eventually integrating them from multiple sources, and the status of the development. The last row of Tables 2.1, 2.2 is dedicated to VisualBib.
Some general considerations are possible:

- the majority of them emerged some years ago, and is no longer under active *development*: the only two active projects are CiteSpace [42] and VOSviewer [95], two freely available domain visualization tools for analyzing emerging trends and changes in scientific literature;

- none of them is a *real-time* application, in the sense that they work on pre-built datasets, or allow the user to upload limited datasets obtained from WOS or other repositories; only VOSviewer can download data through API (i.e., CrossrefAPI, Europe PMCAPI, and several others);

- the possibility of *data integration* from multiple sources is not supported (only partially by VOSviewer); the tools accept specific input formats (for example, datasets generated by WOS, or by Scopus, or created manually);

- the majority of them have been conceived not as Web *applications* but as standalone applications, so requiring the download and the installation.

In experiments performed by Klein et al. [72], they observed that "switching between completely different visualizations confuse the users"; the variety of views proposed by several tools becomes a limitation for their usability; some tools, among them, BiblioViz [91] limits the possible views to only table and network 2D/3D views of bibliographic data; VisualBib uses a unique comprehensive, holistic view, while the major part of them propose several, also very different, views. PaperLens [75] tightly couples views across papers, authors and references in order to empathize the popularity of a topic, the degree of separation of authors and the most cited papers/authors; CiteWiz [62] deals with authors, citations and metadata, features three different views; PaperCube [37], an evolution of CircleView, offers a suite of alternative visualizations based on graphic, hierarchy, and timeline structures. Cybis [53] uses a visive metaphor representing both papers and terms in a cylinder located in the 3D euclidean space. The genealogy of citation patterns, Citeology [81] connects the titles of papers organized in a chronological layout, using a generalized fisheye view; PivotPaths [61] uses a graph representation of authors, publications, and keywords, all integrated in an attractive interface with smooth animations; the demo available online works on a limited dataset of papers in the fields of HCI, information retrieval, and visualization (up to 2012). CitNetExplorer [93] allows visualization of citation networks, offering expansion and reduction operations and clustering of publications. Since most relevant features of VisualBib are the opportunity for the users to manage bibliographies, creating, updating, and editing them, exploring cited/citing papers, exporting and sharing them, Table 2.2 considers these aspects.

We noted that all these tools do not allow users to dynamically choose the list of papers to insert in the bibliography; it is automatically generated applying filter, search or other similar mechanisms on the datasets. The editing of the bibliography, for example, identifying and merging duplicate entries for a same author is only possible editing the dataset; the list of cited/citing papers is often browsable, but only limited modalities exist to choose each of them and extend the set of papers in the current view; only three tools [42, 95, 93] enable users to export and only one [93] to share their bibliographies.

Comparison with ten bibliographic indexes We consider the presence of graphical views in existing bibliographic indexes, which we will analyze in details, on a different perspective, in Section 2.3 with the aim to use them as possible sources of metadata for

Applications	Managing bibliographies				
	Updating	Editing	Cited/citing	Exporting	Sharing
CiteSpace	Static	No	Partially	Yes	No
PaperLens	Static	No	Partially	No	No
Biblioviz	Static	No	Partially	No	No
CiteWiz	Static	No	Partially	No	No
PaperLens	Static	No	Partially	No	No
VOSviewer	Static	No	Partially	Yes	No
PaperCube	Static	No	Partially	No	No
CyBis	Static	No	No	No	No
Citeology	Static	No	Partially	No	No
PivotPaths	Static	No	No	No	No
CitNetExplorer	Static	No	Partially	Yes	Yes
VisualBib	Dynamic	Yes	Yes	Yes	Yes

Table 2.2: Comparing features to manage a bibliography.

VisualBib.
Some of them begin to propose some visual representation, mainly relatively to some metrics (such as h-index, for example). Unfortunately, they do not offer comprehensive and general views on a bibliography, but visualize single metrics; neither the opportunity to compare two or more authors at run-time: AMiner [1] is (together WOS, see below) the most graph rich: it shows some author statistics in a radar diagram, the research interests in a river diagram, all the co-authors in a star graph, and the scholar's trajectory in a map.
Google Scholar [17] shows, chosen an author, the histogram of citations over years; Microsoft Academic [18] uses a similar histogram, adding also the number of papers; Scopus visualizes for the authors their h-index graph, a histogram for citations, and various pie/line charts related to their production (documents by sources, by type, by year, by subject). Scopus [23], WOS [24] and Google Scholar are the only which allow users to create lists of selected papers: Scopus offers for this specific bibliography pie/line charts, while WOS proposes for 16 different components (like WOS Categories, Publication Years, etc.) 16 different tree (or alternatively bar) diagrams, and a set of separated histograms to describe so-called citation report. CiteSeerX [3], CrossRef [5], DBLP [9], OpenAire [19], OpenCitations [20], and Orcid [21] do not present visual representations of (meta)data.

In summary: considering the specific tools, the active projects are few; none of them uses "live" repositories; few tools enable users to dynamically update, edit, export and share their bibliographies; the interfaces are not always usable and the system are not Web applications; considering the existing bibliographic indexes, they not offer interesting visual support for a researcher, interested in the creation, management and sharing of a bibliography.

2.2 Basic functionalities and some screenshots of VisualBib

VisualBib is an online application, freely available for research and teaching, not for commercial purposes.

The first prototype has been released in September 30, 2017; the version 1.0 in February 15, 2018 and version 2.0 in September 1, 2018. In the version 2.2[1], described in this section, VisualBib retrieves data in real-time from the Scopus, OpenCitations and CrossRef/Orcid repositories. For querying Scopus, being a commercial service, it is necessary to navigate in VisualBib from a subscriber's domain in order to get the required data from the Scopus API. A more sophisticated version (3.0) will be presented and evaluated in Section 2.9. The data providers available in the current version were chosen after evaluating the eligibility of the metadata provided by various data sources (as illustrated in next Section 2.3).

A user can create or enrich a bibliography in various ways:

1. searching for a paper, by its DOI (Digital Object Identifier), or, in Scopus, also by Scopus id;

2. searching for an author by nominative, or ORCiD (Open Researcher and Contributor ID), or, by the identifiers applied by the chosen index, in order to retrieve the list of his/her papers and then by selecting what to import;

3. uploading a set of references in .bib format;

4. starting from a paper in the bibliography, retrieve and explore its cited/citing papers.

In the first two cases, user starts choosing one of the 3 available indexes (in Figure 2.1, top-left, the selected index is Scopus), and fills the appropriate fields of the form; in the third case, the user selects the button `Import BibTeX` and upload a file in BibTeX format; in the last case, the user explores the cited/citing papers, following the procedure discussed below and illustrated in the Figure 2.2.

The result will be the visualization of the bibliography in a narrative view. Figure 2.1, bottom displays the narrative view generated by searching the name of a given author ("Dattolo A."): in order to disambiguate between homonyms, the list of 4 found authors is enriched with their name, and, if present, ORCiD, affiliation, subject areas, and OpenCitations's id. Once the user selects the author, the system will fetch the list of the publications, showing a progress bar and an estimation of the residual loading time. In the case of long lists, a stop button is displayed to allow user to interrupt the loading process and examine the partial set of retrieved data. A form containing the ordered list of found papers is then presented to the user who can choose the subset of the publications to import in the narrative diagram, shown at the bottom of Figure 2.1.

The narrative view (see next Section 2.5) is a 2-dimensional space: the horizontal dimension is the time, discretized by years; the vertical dimension is spatial and is used to properly organize authors' names, papers and their relationships. The diagram includes the last names of the authors involved in at least one paper of the current set: each author is associated with a goldenrod line that connects all his/her papers, from the oldest to the newest, giving an indication of the author's professional path (clearly limited to the set of imported publications) over the years. The papers are represented by colored, round-cornered square items (in blue and green in Figure 2.1). Following the second column of the legend of Figure 2.1, on the bottom-left:

[1]The detailed phases of the app development are available online `http://visualbib.uniud.it/en/development/`

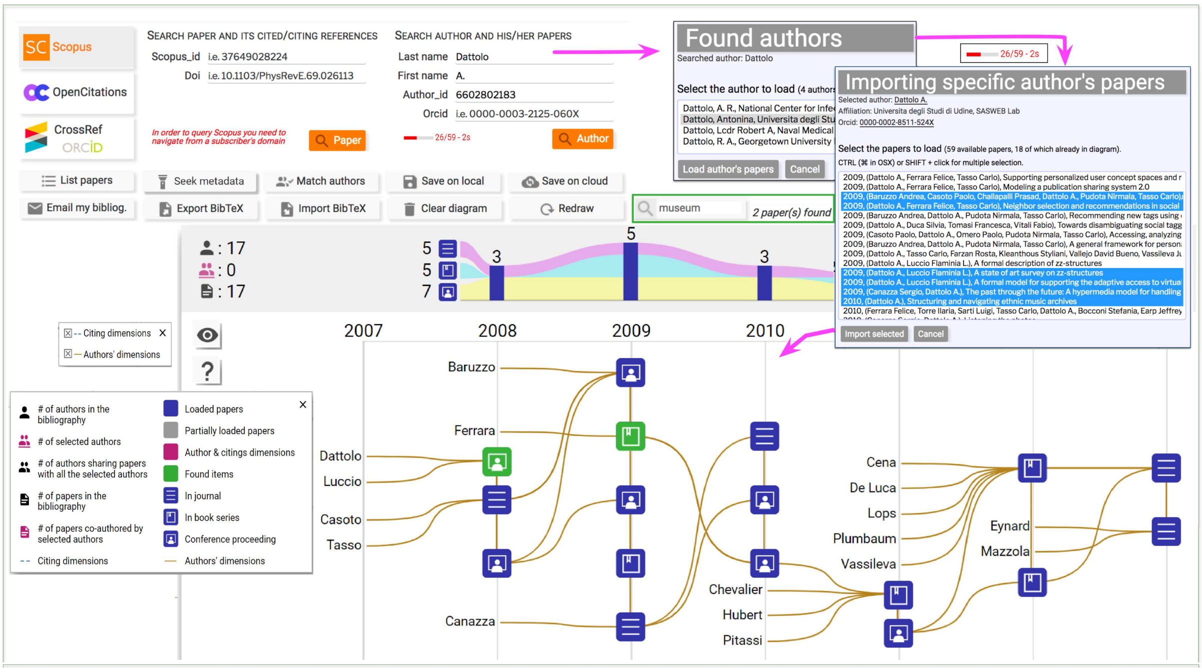

Figure 2.1: The main interface of VisualBib.

- the three different icons associated to publications distinguish between *journal papers*; *books* or *book chapters*; *conference* or *workshop proceedings*. If the type is different or unknown, an empty icon is associated to the item;
- the color of the icons indicates the paper's state: *blue* is associated to a completely loaded paper (all the needed data and metadata have been loaded); *gray* indicates a partially loaded paper, which has been retrieved during a cited/citing search (this operation returns only a subset of paper's metadata); *magenta* is used to emphasize semantic relationships during user interaction, as described later; and, finally, *green* marks the papers found by means of a textual search (in Figure 2.1, they are the papers found looking for "museum" - 2 papers found).

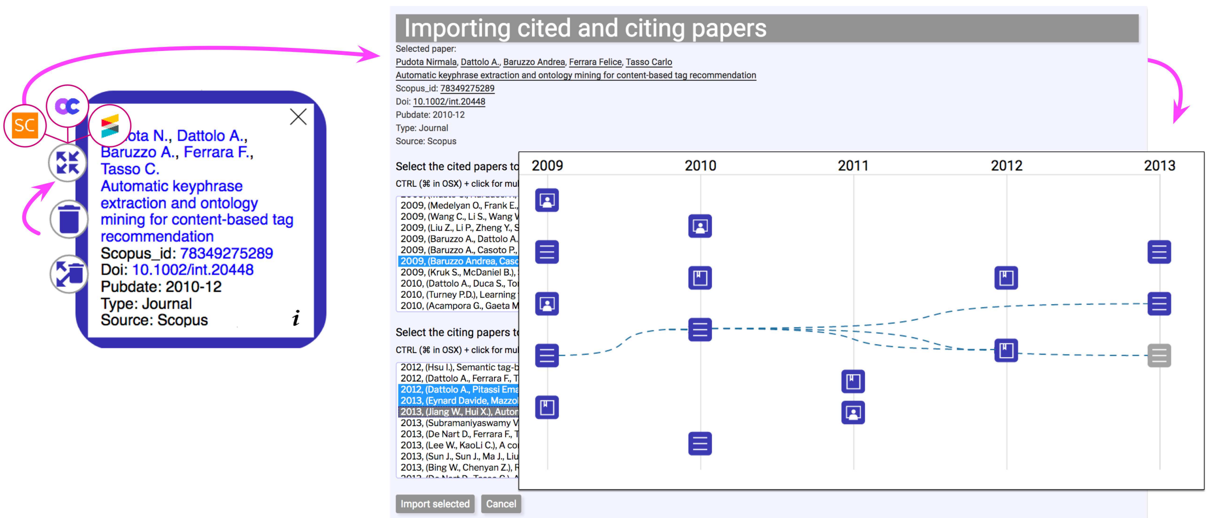

Figure 2.2: Adding cited/citing relationships.

2.2.1 Visual analytics and information discovery

VisualBib enables users to accomplish some visual analysis tasks on the current bibliography by interacting with the narrative view diagram. In the following we illustrate how our tool supports the extraction of bibliographic data and the exploration of the relationships and the citations network.

Distribution of papers and publication types over time The histogram above the timeline in Figure 2.1 shows the distribution of the papers by year of publication, while the overlying stacked area chart, the frequencies of the three publication types: moving the mouse over the colored areas or on the three (publication types) icons, the histogram changes for showing the corresponding counters.

Paper metadata Clicking over a paper icon, a pop-up window (see Figure 2.2, top-left) shows its bibliographic data, where the authors' names, the title, the id and the DOI of the paper are links towards dedicated Web pages. An extended description is accessible by clicking on the *"i"* icon: the amount of metadata presented depends on the source index and can be integrated, querying the Scopus API through the *"Seek metadata"* feature, discussed later.

Citation network A click on the four-arrows icon (see Figure 2.2, top-left) and the next choice of the index where to search citations (in our example Scopus), loads, in a separate form, the list of cited/citing papers (Figure 2.2, center) found on the selected data source. Within each selection list, containing respectively all the cited/citing papers, each item already in the diagram, appears highlighted in blue. Users may select, from the two lists, the documents of their interest and import them (with the related relations) in the diagram. In the example of Figure 2.2, the user chooses to import the three pre-selected papers plus a new one and, as shown on the right of the same Figure 2.2 the cited/citing relations become visible as blue dashed lines.

The production of an author and some collaboration metrics By clicking over an author label (Figure 2.3, left), the application emphasizes in magenta the path connecting all his/her papers in the current bibliography, from the oldest to the newest. The number

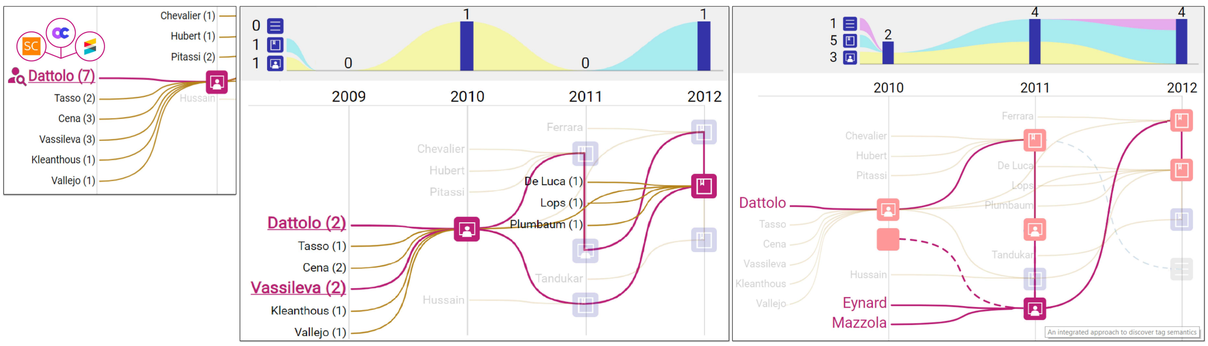

Figure 2.3: Visual analytics of collaboration networks and views on the papers related to a given selection.

in brackets, appended to the author's labels, indicates, for the selected author, the number of papers in the bibliography and, for all the other authors, the number of publications in common with the first selected one. The authors without collaborations with him/her are temporarily obscured by applying partial transparency to the corresponding labels, as shown in Figure 2.3, left, for the author placed under the paper icon. Furthermore, when we select an author, a search icon appears on the left of the name making possible to search the papers available on a specific index (the three icons which appear on the name Dattolo in Figure 2.3, left).

Co-authored papers and multiple collaborations When users select more than one author, by clicking on each name (Figure 2.3, center) all the papers in common between the selected authors are emphasized in magenta and the number of papers written in collaboration with the current set of the selected authors is reported near each name. Also in this case the authors without collaborations with the selected ones are temporarily obscured. When an author is selected, or deselected by clicking again on the label, all collaboration counters, the histogram and the area chart are updated to reflect the current set of co-authored papers.

Related papers Finally, by moving the cursor over a paper icon, the narrative view highlights the network of cited/citing papers and the paths related to the co-authors and their papers (related papers). The focus paper is emphasized in magenta while a salmon color is applied to related papers (Figure 2.3, right).

2.2.2 Other features

In addition to visual analysis of bibliographies, VisualBib offers a dynamic environment to progressively expand and refine a them through some features we illustrate in the following.

List papers To generate the list of the references of all the loaded papers, in textual format.

Seek metadata To check for the availability of additional metadata for each paper in the bibliography imported from Scopus comprising the list of assigned keywords, the abstract, the citation count, the open access flag, the publication name, the publisher and others.

Match authors Since VisualBib retrieves papers and authors metadata from multiple providers, the automatic matching of the same author is only applied if a univocal ORCiD code is available, as explained in Section 2.3. The `Match authors` feature offers users a means to check duplicated author names by comparing their respective publications in order to determine whether they are the same person or not. In this case, the user can apply a wizard in order to merge redundant data.

Save on local It performs the saving of the current bibliography into the local storage of the browser. At the next load of the application page, if a bibliography is available in the local storage, the system will ask the user whether to reload or ignore it.

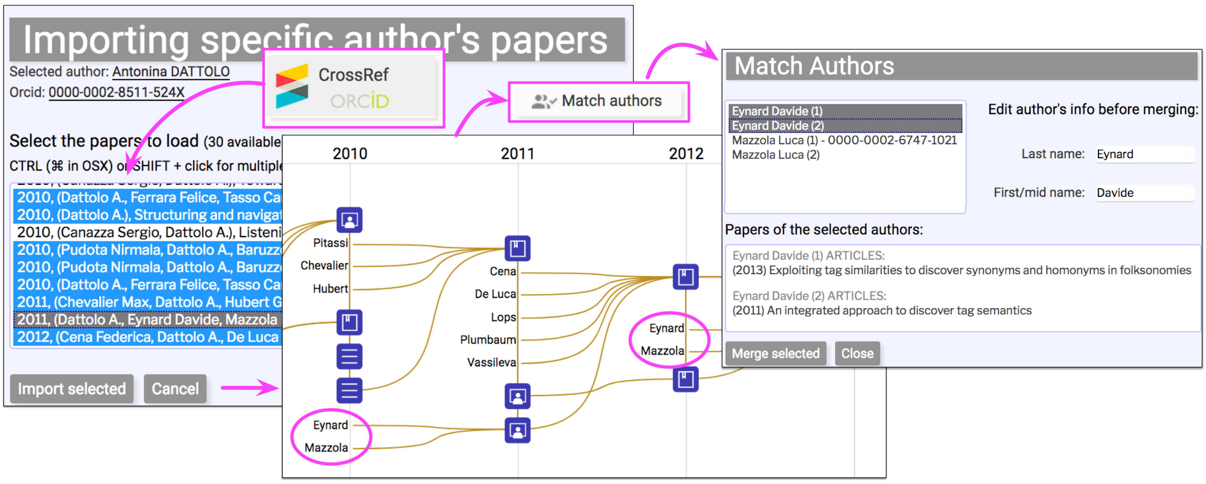

Figure 2.4: The `Match authors` wizard.

Save on cloud It offers users the possibility to save the bibliography in a remote database. A saved bibliography can be accessed from anywhere if provided by a specific link: the user can share the bibliography in read-only (or in read-write) mode with students or colleagues. VisualBib generates also an embed code which allows users to embody a bibliography, in the narrative view format, into an external Web page.

Email my bibliographies Entering an email address, the list of the owned bibliographies, including all the read-write and read-only links will be sent to the user. In the message user will also find the links to permanently delete any of the listed bibliographies.

Export/Import BibTeX It allows users to automatically export (resp. import) a bibliography in the .bib format. In the BibTeX file produced by export, some not standard fields are added to encode the authors and the citation information, necessary to recover, during a next import, all the relations between items.

2.3 Data analysis

In previous Section 2.1, we analyzed the lacking of graphical components in ten, well-known, bibliographic indexes; since they index millions of papers, authors and related metadata, they represent for us the real-time datasets from which potentially extract the metadata necessary for using our tool. For this reason, in Table 2.3, we describe the relationship mining between the user tasks, supported by VisualBib, the visual interactions and the user cognitive processes, and the possible metadata extraction from the ten cited indexes. Successively, we refine the *specifications*, required by VisualBib in order to get, merge and connect bibliographic metadata from these indexes, with the aim to *identify* which of these ten indexes can provide, in real-time, the metadata necessary to VisualBib. Finally we discuss some *open issues*.

In the Table 2.3, applying the analytic task taxonomy in information visualization proposed in [35], we list, in the first column, the low-level data analysis tasks, supported in VisualBib; in the second, the corresponding user visual interactions and, in the third, the corresponding items in the taxonomy. Then, for each user task, we indicate if it is supported in ten of the major bibliographic indexes. We note that, excluding Scopus and

VisualBib' user tasks	Visual interactions	Taxonomy	AMiner	CiteSeerX	CrossRef	DBLP	Google Scholar	Microsoft Academic	OpenAire	Open Citations	Scopus	WOS
Find papers metadata(i.e. title, authors, DOI, etc.)	- mouse over papers - click on papers - list papers	Retrieve value	●	●	●	●	●	●	●	●	●	●
Search / import papers from data sources	- search for paper - search for author - search for citations & references	Filter	◐(a)	●	◐(a)	◐(a)	●	●	◐(a)	◐	●	●
Find papers of an author	- author selection	Filter	●	●	●	●	●	●	●	●	●	●
Get co-authored papers	- multiple authors selection	Filter, Cluster	○	○	○	●	○	○	○	○	●	●
View the collaboration network	- author selection - multiple authors selection	Filter, Cluster	●	○	○	●	○	●	○	○	●	●
Find papers by keywords	- local search	Filter, Cluster	●	●	●	●	●	●	●	◐	●	●
Get summary data (b)	- author selection - multiple authors selection - data summary - list papers	Compute derived value	●	●	○	◐	●	●	◐	○	●	●
Get # of papers per year	- examine histogram	Compute derived value	●	●	●	●	●	●	◐	○	●	●
Order papers by year / get most productive year	- examine histogram - examine timeline - author selection	Sort, Find extremum	●	◐	●	◐	●	◐	◐	◐	●	●
View document types distribution	- examine histogram - interact with area chart	Characterize distribution	○	○	○	○	○	○	○	○	●	●
Determine timespan of a bibliography	- examine timeline	Determine range	●	◐	◐	●	◐	◐	◐	◐	●	●
Find duplicate authors	- match authors	Find anomalies	○	○	○	○	○	○	○	○	○	○
Get related papers	- mouse over a paper icon	Cluster	●	◐	○	○	○	●	●	○	●	●

Legend: ● full provided; ◐ partially provided; ○ not provided
(a) Citations or references not provided (b) Number of papers, authors, co-authored papers, collaborations, citations, etc.

Table 2.3: List of the low-level data analysis tasks supported by VisualBib compared with those provided by a set of well-known bibliographic indexes.

WOS, the other indexes support, in a sparse way or partially, these user tasks. At this level of analysis, we considered the functionalities usable directly by the Web platform of these indexes, without considering that some metadata or tasks could be performed combining metadata, provided, for example, by API; a limiting case is OpenCitations, that only manages simple search operations, but provides an interface for executing SPARQL queries.

Specifications In order to investigate the feasibility of integrating in VisualBib various bibliographic indexes as real-time data sources, we individuate the following list of specifications: **Paper's metadata by DOI** - starting from the DOI of a publication, VisualBib needs to retrieve metadata such as the title, the publication year, the list of co-authors (possibly complete of unique identifiers), the type of publication (i.e., conference paper, journal paper, book or book chapter, etc.) and, possibly, other metadata such as keywords, external links, abstract, metrics; **Citations by DOI** - starting from the DOI of a publication, VisualBib would like to retrieve the lists of cited and citing papers, possibly complete with the DOIs and the lists of uniquely identified authors;

Authors by name - starting from the name/surname of an author, VisualBib needs to retrieve the list of the corresponding authors, enriched with their *ORCiD (Open Researcher and Contributor ID)* and/or internal unique identifier; **Author's papers by ORCiD** - starting from an ORCiD or another author unique identifier, VisualBib needs to retrieve metadata related to the current affiliation, subject areas, list of the publications, complete of DOIs and lists of authors, possibly uniquely identified.

Bibliographic indexes' APIs analysis Table 2.4 summarizes, for each platform, the availability of specific data retrieval functions with reference to the above specifications. The evaluations have been formulated analyzing the online available documentation, in some cases rather incomplete, and, where possible, directly testing the services.

Platforms	Paper Metadata (by DOI)				Citations (by DOI)		Authors (by name)		Papers (by ORCiD)		
	title	year	auth.	auth. ids	cited	citing	ORCiD - local ID	aff.	aff. - area	list of papers	list of authors
AMiner	?	?	?	?	?	?	?	?	?	?	?
CiteSeerX	○	○	○	○	○	○	○	○	○	○	○
CrossRef	●	●	●	◐	◐	○	○	○	○	●	◐
Orcid	○	○	○	○	○	○	●	●	●	●	○
DBLP	○	○	○	○	○	○	◐	○	○	○	○
Google Scholar	○	○	○	○	○	○	○	○	○	○	○
MS Academic	●	●	●	●	●	?	●	●	◐	◐	◐
OpenAire	●	●	●	○	◐	○	◐	○	○	○	○
OpenCitations	●	●	●	●	●	●	◐	●	●	●	●
Scopus	●	●	●	●	●	●	●	●	●	●	●
WOS	●	●	●	●	●	●	●	●	●	●	●
WOS Lite	●	●	●	●	○	○	●	?	?	?	?

Legend: ● full provided; ◐ partially provided; ○ not provided; ? not well documented, to be verified.

Table 2.4: Data provided through API services.

Although AMiner provides comprehensive search and mining services for researcher social networks, and an API service exists [7], at the moment the documentation is not adequate to use it. At the moment, neither CiteSeerX, neither Google Scholar provide an API service for supporting third-party applications, while CrossRef and Orcid appear

be almost complementary: in fact, the API Rest services [6] of CrossRef retrieve the metadata of a publication given its DOI and, if available, get the references to the cited papers but for the citing ones. It is also possible to retrieve the list of publications of an author given the ORCiD but not to search by name/surname. On the other hand, Orcid provides search service of authors that have registered an ORCiD code: in this case it is possible to retrieve their metadata and a list (possibly incomplete) of their publications in the form of collections of DOIs. Orcid APIs do not provide any service to retrieve metadata about papers.
DBLP offers an API service for the research of authors and publications. Unfortunately the service is based on textual queries and the DOI or ORCiD identifiers cannot be used as search keys although these information are often provided in the responses. This fact, together with the lack of data concerning the cited / citing documents, makes its integration in VisualBib difficult.
According to the documentation, the integration with VisualBib of Microsoft Academic could be feasible through the *Evaluate* method of *Paper* and *Author* entities although some significant fields, like ORCiD and citing references, appear to be absent. We plan to verify the feasibility of an effective interfacing with the Microsoft Academic APIs in the future work.
The current API version of OpenAire offers the retrieval of publications metadata but provides limited features for author search and disambiguation. However, a forthcoming API release has been announced with improvements; we will analyze the new services in the future work.
OpenCitations provides a REST API service to query the internal corpus. For each data retrieval task, VisualBib prepares and submits a list of specific SPARQL queries to the single OCC API endpoint and extracts the needed metadata from the JSON responses.
Scopus offers a rich set of API [13] to retrieve authors and papers metadata, including references and citations. The API services of Scopus require that the HTTP API calls must originate from an IP address inside the domain of a subscriber organization, in order to be processed. For subscribers, the data provided by Scopus API are complete, returning rich metadata about papers, authors, references and citing documents.
Also WOS [24] provides a rich set of task based APIs to query more than 70 million records. The recently published WOS API Expanded is a commercial service that could provide all the VisualBib's needed data. A problem arises from the mechanism used to authorize applications to access the indexes, based on an API key. In absence of other protection mechanisms, the integration of the WOS API in VisualBib, would result in a exposure of WOS's data to not subscribers users, in contrast to their data policy. For this reason, now we are exploring possible alternative technical solutions that are compatible with the terms of use of the service. Unfortunately, the service offered by WOS API Lite returns only a restricted subset of WOS metadata: for example, all cited/citing references are not provided.

Based on this analysis, we decided to implement at first the procedures to query and retrieve data from Scopus and OpenCitations as these fully meet all the specifications; then, observing the complementarity of the data provided by CrossRef and Orcid, we decided to use them in combination, performing authors searches on Orcid and papers metadata retrieval from CrossRef.

Issues concerning multiple data sources Many issues may arise during the retrieval of metadata about specific papers or authors: depending on the data source, some significant information may be missing or supplied in a different form. For example many papers and authors are not provided with universal identifiers such DOI or ORCiD, or are marked only by a local identifiers, e.g. Scopus id, applicable exclusively within a specific domain. In order to match and merge data from multiple sources, these issues must be managed defining a strategy for dealing with the various cases. A detailed analysis of these issues and a description of the strategies adopted to deal with are presented in next Subsection 2.7.4.

2.4 Modeling a visual bibliography by zz-structures

In this section we formally introduce a model of Visual bibliography based on zz-structure and propose a new zz-view, called *narrative view*, conceived to support the exploration the underlying bibliography zz-structure.

2.4.1 Zz-structure model in VisualBib

A bibliography can be thought of as a network of authors and papers, interconnected by citing/cited dimensions and containing, for each paper, details for associating to it *authorship*, *title*, *DOI*, *editorial collocation*, *repositories* on which it is indexed, etc..

Definition 2.4.1. Visual Bibliography - A visual bibliography VB is as a zz-structure, where

- $V = \{A, P, DE, \mathring{P}C, \mathring{C}P\}$, the finite set of vertices, is composed by:
  - a finite set of authors A, papers P, and details DE, associated to the papers. DE is constituted by composite cells;
  - $\mathring{\text{P}}$C and $\mathring{\text{C}}$P are compound cells, containing, for each paper:
    - the set of PC - *Papers Cited* by it;
    - the set of its CP - its *Citing Papers*.
- $D = \{d.a_1, \ldots, d.a_n, d.time, d.cited, d.citing, d.details, d.author\text{-}l, d.title\text{-}l, d.ID\text{-}l, d.doi\text{-}l\}$, where:
  - $d.a_1, \ldots, d.a_n$ identify the dimensions, which group the publications of the authors $a_1, \ldots a_n$; each of these last is the maincell of each dimensions;
  - *d.time* is constituted by the parallel ranks $d.t_1$, …, $d.t_m$, which group the papers of the bibliography published during each year $t_1, \ldots, t_m$; these time marks are the maincells of each rank;
  - *d.cited* and *d.citing* respectively connect each paper to the sets of the papers cited by it and that cite it; these two dimensions are constituted by parallel ranks, one for each paper;
  - *d.details* is constituted by parallel ranks. Each paper is associated to a composite cell, containing specific details, which link to external information:
    - *d.author-l*: this dimension is constituted by parallel ranks: each author is connected to a related Web page on the repository which the paper has been retrieved from (Scopus, OpenCitations or CrossRef/ORCiD);

  * *d.title-l* and *d.ID-l* link to the paper's Web page on the repository, respectively using as access key the title or the ID of the paper in the specific repository;
  * *d.doi-l* links to the DOI Web page of the paper.

Furthermore, VB becomes operational thanks the two sets of views and mechanisms, introduced in next Section 2.5.

Figure 2.5 proposes an example of bibliography, represented in terms of a zz-structure. The zz-cells of the bibliography are 4 authors $\{a_1, \ldots, a_4\}$, 8 papers $\{p_1, \ldots, p_8\}$, and 2

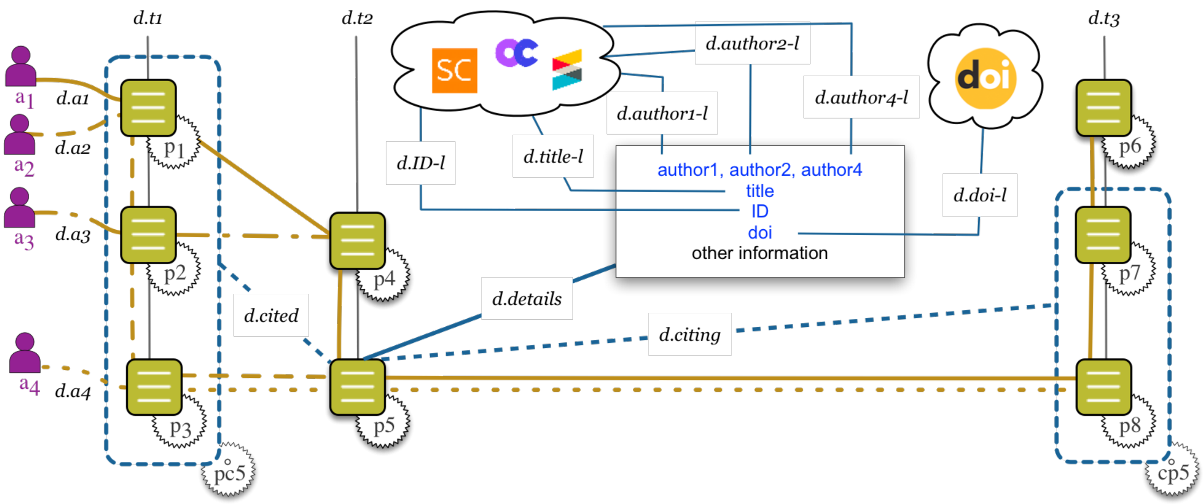

Figure 2.5: An example of bibliography represented using a zz-structure.

compound cells $\{\mathring{pc}_5, \mathring{cp}_5\}$, composed by subsets of papers, respectively cited by/citing the paper p_5. Following Figure 2.5, we analyze the involved dimensions:

- $a_1, \ldots a_4$ represent the maincells of the four dimensions $d.a_1, \ldots d.a_4$; each of them groups the publications of the related author. For example, the dimension of the author a_1 is composed by $(a_1, p_1, p_4, p_5, p_8, p_7, p_6)$;

- $d.t_1, d.t_2, d.t_3$ represent the three parallel ranks related to the papers published in the years t_1, t_2, and t_3;

- *d.cited* connects the paper p_5 to the compound cell $\mathring{pc}_5$, containing the list of its references (p_1, p_2, p_3), while *d.citing* connects p_5 to $\mathring{cp}_5$, containing the papers (p_7, p_8), which cite p_5;

- *d.details*: p_5 is associated to a composite cell, containing its details, which link to external information:

  – $d.author_1$-l, $d.author_2$-l, and $d.author_4$-l connect the 3 authors a_1, a_2, and a_4 to their Web pages on the repository which the papers have been retrieved from;

  – *d.title-l* and *d.ID-l* link to the paper's Web page on the repository;

  – *d.doi-l* links to the DOI Web page of the paper.

Besides these main dimensions, the model owns a wider potential, provided by the possibility to identify new ways to semantically connect and visualize relations among authors, papers, and related metadata. For instance, papers might be related using "subject areas", "keywords", "users' tags" dimensions, generating customized narrative views based on personal labeling of papers.

2.5 Zz-views in VisualBib

Starting from the VisualBib zz-structure model, in order to find an effective representation that takes into account the temporal, author and citation dimensions of a bibliography, we proposed two original zz-views called *deep-view* and *narrative-view*. In this section we describe the specific measures needed to obtain comprehensible and interactive views in VisualBib starting from the formal definitions in Sections 1.3.4 and 1.3.5.

2.5.1 Deep views

In the context of visualbib, we adopted the *deep-view* model to visually represent the citation relationships between papers in the bibliography. Specifically, a couple of *deep-views*, centered on a paper p_i and applied respectively to *d.cited* and *d.citing* dimensions, generates a series of direct connections, rendered through blue dashed lines, from p_i to all its cited papers belonging to the $\mathring{p}c_i$ compound cell and citing papers in the compound cell $\mathring{c}p_i$.
An example of deep-view is shown in Figure 2.6: on the left we consider an extract of the

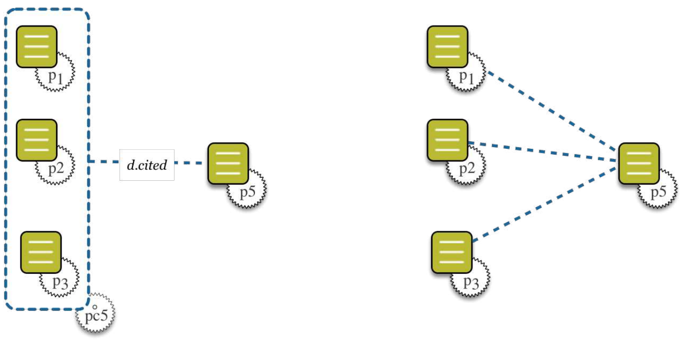

Figure 2.6: p_5 is connected with $\mathring{p}c_5$ by *d.cited* (left); the deep view explodes the connections (right).

zz-structure of the Figure 2.5, where $\mathring{p}c_5$ and p_5 are connected by the dimension *d.cited*. On the right, is displayed the deep-view, where the citation link is exploded in three links.

All the *deep-views* associated to each paper in the bibliography become visible when user activates the check box labeled as *citing dimension* (see Figure 2.1). In order to improve the readability of the diagram, user may highlight the *deep-views* related to a specific paper by moving the mouse over it: the involved papers and links are colored red and all the others are temporarily darkened. Figure 2.7-top shows the complete citation network of a small bibliography and the highlight of the two deep-views centered on a specific paper, showing respectively the *d.cited* and *d.citing* relationships.

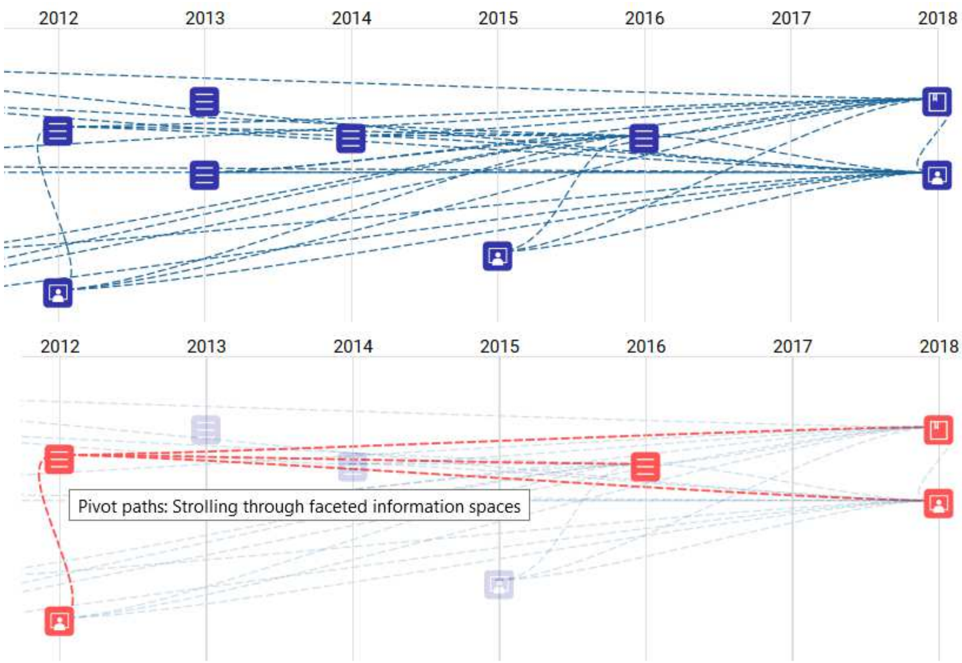

Figure 2.7: The complete citation network (top) of a small bibliography and the highlight of the two deep-views associated to the first paper of 2012 (bottom).

2.5.2 Narrative views

The main view of VisualBib consists of a *narrative-view*, formally defined in Section 1.3.5. The *narrative-view* can be overlayed by a series of *deep-views* showing the citation relationships between papers. As previously mentioned, the layer related to the citation dimensions can be hidden as well as the visualization of the authorial dimensions. Applying the Definition 1.3.3 of *narrative-view* to VisualBib, we instantiate the set of items P to the *papers*, the set of agents A to the *authors* of papers and the temporal marks to the *years* of the publications.

An example of narrative view, is shown in Figure 2.1.

2.5.3 Topological constraints in the narrative view

Starting from definition 1.3.3, we specifically describe below some topological constraints in the visualization of the narrative view in VisualBib.

Given the sets of papers $P = \{p_1, \dots, p_{|P|}\}$ and the set of authors $A = \{a_1, \dots, a_{|A|}\}$, we introduce some functions and notations:

- $position : P \cup A \to X \times Y$, where $X \subset \mathbb{N}, Y \subset \mathbb{R}$, is a function to map papers and authors in the bi-dimensional space of the view: the X axis represents the years of the publications, extended, on the left, with an additional year and identified below by the notation $position_x$, while the Y axis defines the spatial dimension where to collocate papers and authors ($position_y$);
- $authors : P \to \mathcal{P}(A)$ returns the set of authors for a given paper p;
- $AuthorPaths = \{(a, p_1, \dots, p_n)\ \forall a \in A : a \in authors(p_i)\ \forall i = 1, \dots, n\}$ is the set of paths that link authors with all their papers. The papers are sorted in ascending

order by year of publication. The path is represented by a solid line and its label is a (the name of the author);

Now we introduce the related topological constraints. The scale of the axes is dynamically computed at every change in the bibliography in order to cover, respectively, the entire temporal span and the maximum number of items per column. The positioning of the items along the vertical dimension is critical for a proper interpretation of the information in the diagram; and a specific algorithm has been designed in order to achieve the following features:

1. no overlapping of the papers published in the same year; fixed a threshold d_{min} for the minimum distance between the papers having in common the publication year should be:

   $|position_y(p_i) - position_y(p_j)| \geq d_{min}$,

   $\forall p_i, p_j \in P$, $p_i \neq p_j$, and $position_x(p_i) = position_x(p_j)$;

2. a balanced distribution of the papers along the vertical dimension; we first identify a central common axis in order to position all papers around it. With this aim, we

   (a) find t_{max}, the year with the maximum number of publications:
   let $P_t = \{p \in P : position_x(p) = t \in T\}$ where $T = \{position_x(p) : p \in P\}$.
   Then $t_{max} : |P_{tmax}| = max|P_t|, \forall t \in T$.

   (b) define the extent of the vertical space, proportionally to the number of publications and the chosen d_{min}, as the open interval:
   $(0, |P_{tmax}| \cdot d_{min})$

   (c) position the central horizontal axis: chose as unity scale of the y axis $d_{min} = 1$; so, the central axis has the equation:
   $y = \frac{|P_{tmax}|}{2}$

   (d) position the papers along the vertical dimension: $\forall t \in T$, and $\forall p_i \in P_t$
   $position_y(p_i) = \frac{|P_{tmax}|-|P_t|-1}{2} + i$ where $i = 1, \ldots, |P_t|$.

   The order of the papers determined by the index i is not significant, any of the $|P_t|!$ permutations of papers of any set P_t is adoptable. In this context, an interesting and open point would be to find a set of permutations of each P_t which minimizes the number of crossings of the authors and citing/cited links that join the papers. The cardinality of the solution space for this problem is $\prod_{t \in T} |P_t|!$

3. the correct positioning of the authors' labels for each authorPath $(a, p_1, \ldots, p_n) \in AuthorPaths$, we position the author's label in

   $position_x(a) = position_x(p_1) - 1$

   For example, in Figure 2.1, the authors' labels of the paper of the 2008 are positioned in the 2007 column;

4. a regular space distribution between papers and authors' labels in every year in order to avoid collisions with other papers and labels. This is achieved by moving eventual paper icons and labels in collision towards the bottom. The constraint 1. is also ensured between labels, but scaled by a constant 0.4, that takes into account

the proportion between the height of the paper icons and the font used for the labels. For this reason, the minimum distance between papers is d_{min}, while the minimum distance between papers and labels, or labels and labels, is $0.4 \cdot d_{min}$.

2.6 Use case scenario

In this section we discuss a use case scenario to show how VisualBib can support an author in the import, refinement and export of a bibliography starting from a set of references provided in a BibTeX file.
Figure 2.8 presents the sequence diagram related to the operation of importing a BibTeX, which starts when users click on the 'Import BibTeX' button, but can be enriched of metadata applying 'Seek metadata' function and refined by 'Match authors' function:

- *Import BibTeX*: preparing and importing in VisualBib a set of references contained in a BibTeX file.

- *Seek metadata*: retrieving extended metadata about all the papers and authors in the bibliography by checking their availability in the Scopus citation index through its API endpoint. Detecting citation relationships between each pair of papers and visualizing them in the narrative diagram.

- *Match authors*: analyzing the list of authors of the papers in order to drive users to identify and merge corresponding entries, checking for any difference between the BibTeX and the Scopus data.

In the following Subsections 2.6.1-2.6.3 we analyze in detail these phases both from a user and system point of view. In the Subsections 2.6.4-2.6.6 we describe, respectively, the refinement of the bibliography by removing and inserting new papers and, finally, the export of the bibliography in the BibTeX format.

2.6.1 Preparing and importing a BibTeX archive

The 'import BibTeX', function analyzes and extracts data from a BibTeX archive: each paper entry in the BibTeX is parsed in order to check the syntax and extract significant metadata, divided into three typologies.

Required metadata for each entry The required fields are *document type*, *title* and *publication year* of the paper.

Additional metadata A set of recognized fields are *author, month, journal, booktitle, publisher, volume, pages, isbn, issn, url, abstract.* Among them, in order to enable user to use all the functionalities of VisualBib, we suggest users to include at least the *author* field, particularly if we do not specify a unique identifier for the paper, for example, *scopusid* (see next paragraph). In the *author* field the names of the authors of the paper must be separated by the 'and' string and specified in the format *last name, name*; in absence of the comma, VisualBib assumes that last word is the author name, and the previous words, the last name.

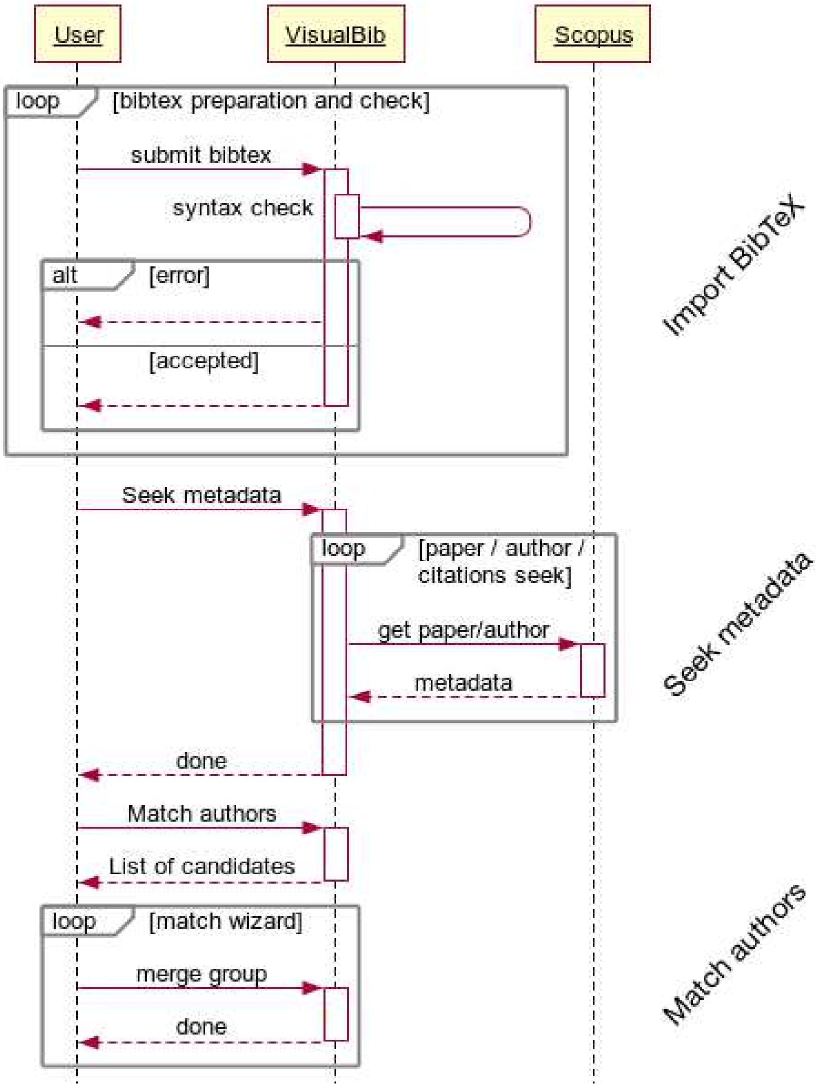

Figure 2.8: Importing BibTeX: use case sequence diagram

Non-standard metadata The system recognizes and acquires a set of non-standard fields like *doi, scopusid, citedby, source, references, keywords, author_keywords, topics* and *authordata*.
We suggest users to include at least the *doi* or *scopusid*; each of them represents a unique identifier of the paper and allows the system to retrieve extended metadata, as described below in Subsection 2.6.2. Furthermore, the presence of this field enables users to explore the citation network of the paper.
These non-standard fields are automatically generated by the *export BibTeX* procedure in order to allow VisualBib to completely reconstruct the bibliography in subsequent import operations. In particular:

- *citedby* identifies the number of citations;
- *source* identifies the provider (among Scopus, OpenCitations and CrossRef/Orcid) from which VisualBib retrieved the metadata;
- *references* identifies the set of cited papers through a comma-separated list of doi or scopusid. If this field is present, VisualBib generates and shows the citations net-

work between the papers in the bibliography, ignoring, in this phase, the references to external documents;

- *keywords* contains the list of semicolon-separated keywords attributed to the paper by the citation indexes: i.e. in Scopus this information is explicitly labelled as *Indexed keywords*;

- *author_keywords* is a list of semicolon-separated keywords proposed by the authors;

- *topics* lists the subject areas associated to the paper;

- *authordata*: BibTeX format does not give the chance to uniquely attribute authors to papers. The names defined in the 'author' field can be ambiguous and/or can be written incorrectly due for example to the presence of diaeresis and accents. Through the *authordata* field, it is possible to specify, for each author, a unique identifier like the *orcid*, the Scopus's *authorId* or, in the absence of these data, a numeric value (which will be used by VisualBib as unique identifier) in order to map him/her to different papers in the bibliography. In the next example, we consider a paper, written by three authors; through *authordata*, we associate the *orcid* identifier (prefix O:) to the first author of the paper, a numeric identifier (prefix I:) to the second author and both the *orcid* and the *authorId* (prefix S:) to the third one.
```
author={Lastname1, Name1 and Lastname2, Name2 and Lastname3, Name3}
authordata={O:orcid-author1,I:id-author2,O:orcid-author3|S:authorid-author3}
```

During the import, for each paper in BibTeX the system will check:

- the presence of the set of required metadata (*document type*, *title* and *publication year*), discarding the paper if missing;

- the presence of the paper in the current bibliography through a *doi* or *scopusid* comparison; if the paper is already present, the new entry will be ignored;

- the presence, for each author of the paper, of a unique *id* in the *authordata* field. If missing, a new author entry will be created in the bibliography: no attempts are made, at this phase, to match existing authors by name. If an author's unique id is provided and it matches with an author in the bibliography, the paper is attributed to him/her without adding new author entries.

At the end of the importing procedure, the *references* field of each paper is examined in order to detect and display the citation relationships within the current bibliography.

Consider a concrete use case; we would like import a BibTeX file[2], also visible in the Appendix A.1, containing an extract of 15 references, cited in the bibliography of this theis. We decided to minimize the effort in the creation of the BibTeX, including only the required metadata, and, if available, *scopusid* or *doi*.

It follows a typical entry, with required metadata and *doi* (or *scopusid*):
```
@article{Federico2017,
title={A Survey on Visual Approaches for Analyzing ...},
year={2017},
doi={10.1109/TVCG.2016.2610422}}
```
For demonstration aims, we included also four entries which identify special cases:

[2]The file used in this example is available online at http://visualbib.uniud.it/biblio-example-in.bib

case 1 - *scopusid* and *doi* are not specified:

```
@article{brooke2013,
title={SUS: a retrospective},
author={Brooke, John},
journal={Journal of usability studies},
volume={8},
number={2},
pages={29--40},
year={2013},
publisher={Usability Professionals' Association}}
```

case 2 - We insert the list of authors for two papers, written by the same authors, but we did not fill the field *authordata* for associating a unique identifier to each of them. Furthermore, in the first paper the order of the authors is incorrect (the first author is Corbatto).

```
@inbook{Corbatto2018vb,
author={Dattolo, Antonina and Corbatto, Marco},
title={A Web application for creating and sharing ...},
year={2018},
scopusid={85058996659}}

@inproceedings{Dattolo2018Vis,
author={Dattolo, Antonina and Corbatto, Marco},
title={VisualBib:Narrative Views for Customized ...},
year={2018},
doi={10.1109/iV.2018.00033}}
```

case 3 - The last name of the second author is misspelled:

```
@article{Chen2010,
author={Chen, C. and Ibek San Juan, F. and Hou, J.},
...
```

The correct last name is Ibekwe-Sanjuan.

As a first action, we upload the BibTeX file using the 'Import BibTeX' button. Visualbib correctly parses the 15 papers and 8 authors, those that we explicitly inserted in the .bib file. The generated narrative view is shown in Figure 2.9. Narrative view and textual bibliography, obtainable by clicking on 'List papers', contain only the data which we included in the .bib file. An example is provided in Figure 2.9-bottom where we opened the metadata preview window related to the single paper in the bibliography published in 2017.
Now we would like to automatically enrich the bibliography; this operation could be performed using the 'Seek metadata' function.

2.6.2 Seek metadata

Seek metadata analyzes the set of papers and authors of the current bibliography checking, for each of them, the availability of extended metadata through the *abstract retrieval* and *author retrieval* Scopus APIs.

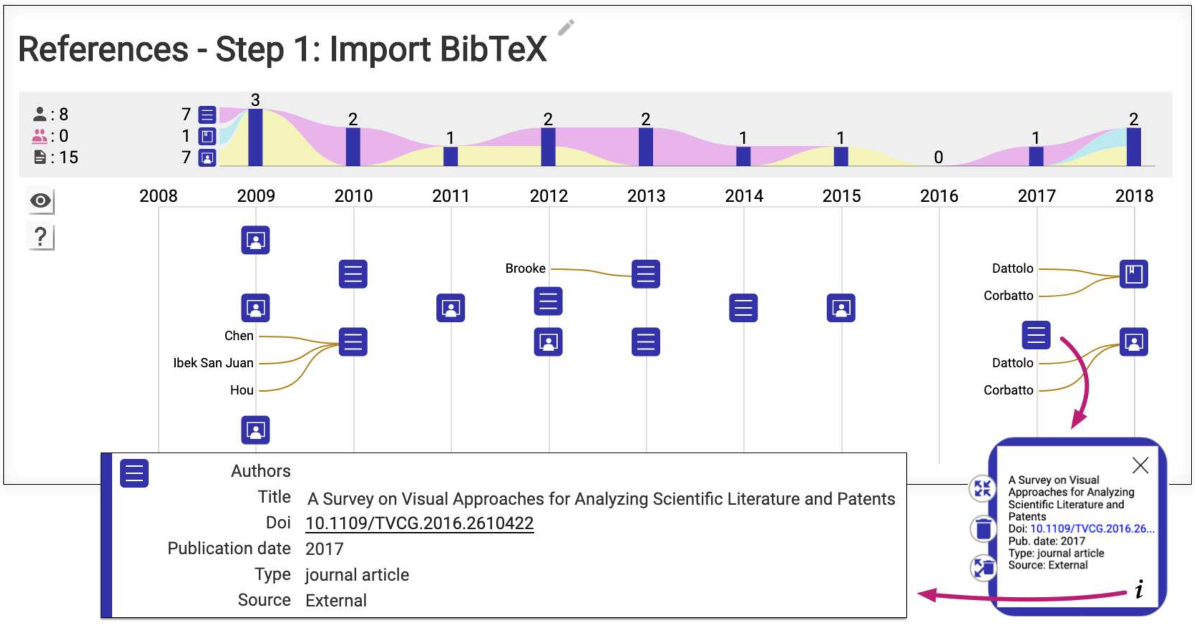

Figure 2.9: The narrative view generated importing the biblio-example.bib listed in Appendix A.1.

The check is performed for all papers having a *doi* or *scopusid* and for all authors including *orcid* or *authorid* identifiers.

- **Metadata of papers**. The metadata retrieved for each paper include, if available, the list of *authors*, *publication year* and *month*, *abstract*, list of *subject areas*, *doi, issn, isbn, publication name, publisher, volume, page numbers, # of citations*, list of *authors' keywords*, *indexed terms*, and the *list of references* of each paper: this last one is also retrieved in order to reconstruct the citation network between all the papers in the current bibliography, and viewable using the 'eye' icon, highlighted in Figure 2.7.

- **Author's metadata**. The metadata retrieved for each author include the *name, last name, orcid, authorid, current affiliation, affiliation history, # of publications, # of coauthors, # of citations, h-index, list of journals (sources), distributions of subject areas (topics)*. All this data are presented in the author details window, shown in Figure 2.11, accessible by clicking on the 'i' icon appearing over the author name.
  During the analysis of a paper, the list of authors retrieved from Scopus is compared with the current authors' list of the papers in the bibliography: authors uniquely identified are automatically matched, otherwise a textual comparison is performed: if the matching is not exact, a new author is introduced and associated to the paper. This procedure covers also the case of an incomplete author list, such as in our example, where we did not insert the major part of authors.
  Possible duplication of authors, due to misspellings in names or to the presence of special characters, could happen in this phase but it will be addressed by the 'Match authors' function described in Subsection 2.6.3.

In order to minimize the number of queries to Scopus APIs, both papers' and authors' metadata are currently cached for a period of 30 days.

Following our example, we apply 'Seek metadata' function, which extends the view of Figure 2.9, enriching the metadata for 12 papers and 36 authors. The procedure adds the *citation dimension* shown in Figure 2.7-top. If we select now the same paper of 2017, considered in Figure 2.9, we can observe in Figure 2.10-left the new, enriched metadata. In Figure 2.10-right, we considered case 1, introduced in Subsection 2.6.1; in this case,

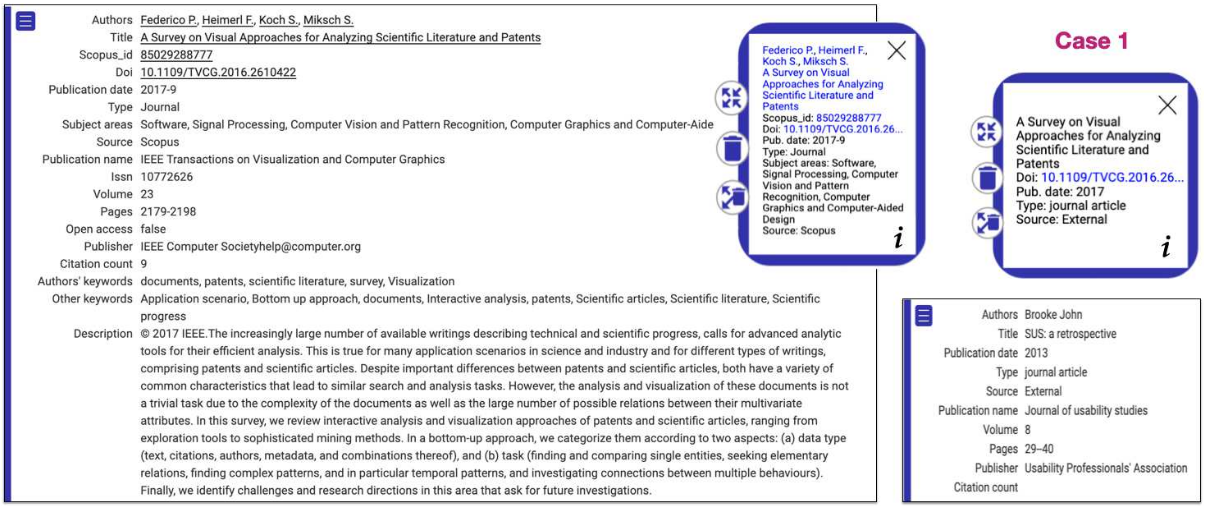

Figure 2.10: Left: Enriched metadata for the same entry considered in Figure 2.11-right: Metadata for case 1, introduced in Subsection 2.6.1.

the fields *scopusid* and *doi* are not specified, and consequently the metadata related to this paper have not been enriched.

Figure 2.11 proposes the new, enriched narrative view, after applying the 'Seek metadata', on the bibliography proposed in Figure 2.9.

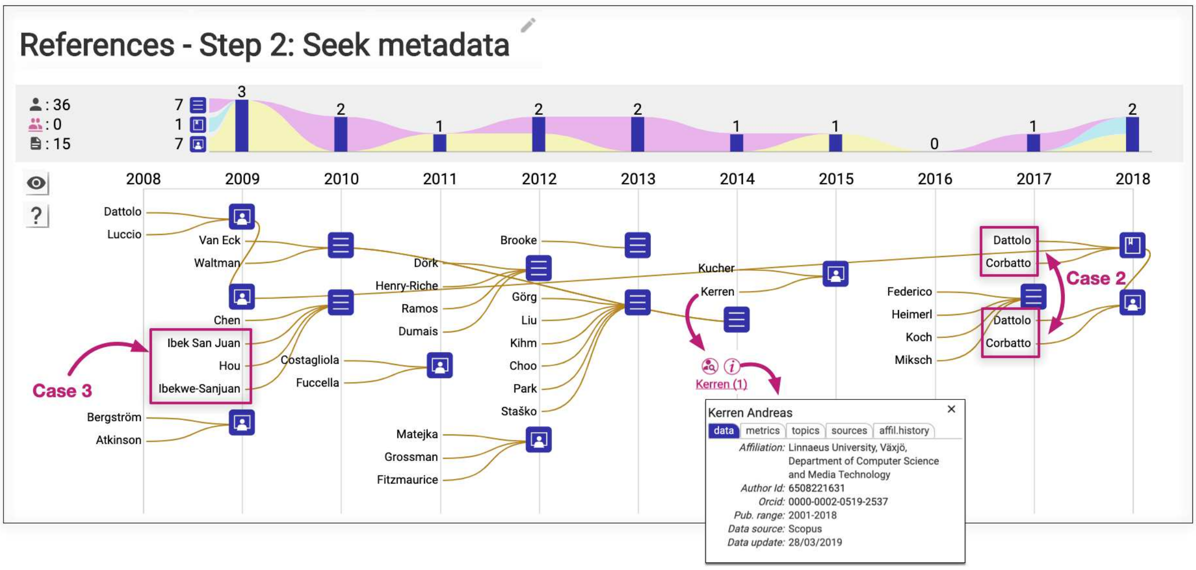

Figure 2.11: The enriched narrative view after the 'Seek metadata'.

It is evident now the presence of 36 authors, instead of the previous 8 (see the counters at the top-left corner of Figure 2.9). Furthermore, for each author it is possible to open the author details window, accessible by clicking on the 'i' icon appearing over the author name. An example is provided in Figure 2.9-center, where we chose the Kerrer author. The author details window shows data, metrics, topics, sources and affiliation history of this author.

In the same Figure 2.11, we highlighted cases 2 and 3, introduced in Subsection 2.6.1. In case 2, we note that the two authors, inserted in the .bib without unique identifiers, have been duplicated. In case 3, the incorrect last name 'Ibek San Juan' is present, but the applied procedure also added the new correct lastname 'Ibekwe-Sanjuan'.

2.6.3 Match authors

The introduced duplication of authors can be checked and corrected through the *Match author* function. Figure 2.12 shows the match authors form which includes: the list of the authors in the bibliography whose last name is present more than once (left); a list of the publications of the selected authors to help users to establish whether or not they belong to the same author (bottom); the proposed last name and first name to associate to the merged entry; the *Merge selected* button to unify, after a confirmation, the selected authors in a single entry; after the merge the system will select the next group of candidate authors; the *Select next group* button to explore next groups without merging the current one. In our case, we use this function and we merge the duplicated/triplicated last names

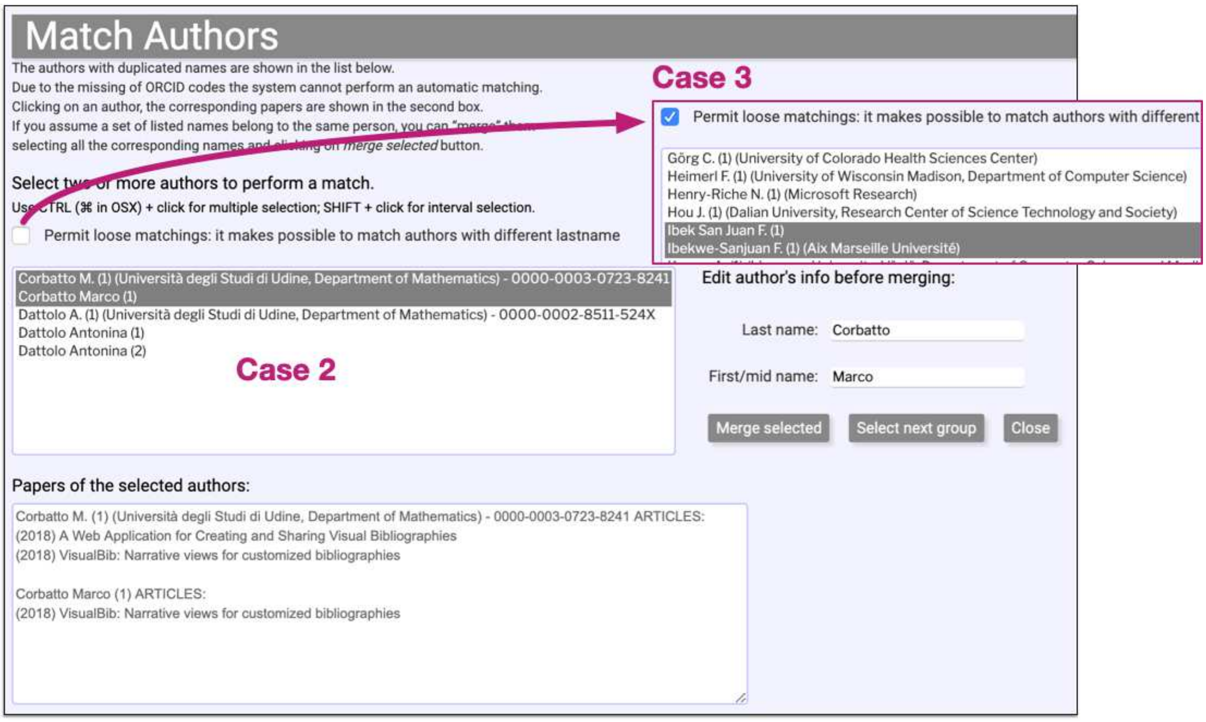

Figure 2.12: The merge authors form.

(case 2), automatically generating the correct order of authors.
A check box, visible above the selection list, permits, when selected, to examine all the authors; it represents in Figure 2.12 the tail end-point of the arrow which leads to case 3. In this case, users can freely select and merge any set of authors. This feature is useful to merge the repeated entries of a same author reported with spelling errors in the name or in the presence of special characters. Using this opportunity, we find the incorrect last name of case 3, and we can apply the merge procedure, choosing the correct last name

or correcting the existing ones. Once merged, all the papers associated to each entry will be connected to the unified author entry.

2.6.4 Refinement: deleting papers

Starting from the enriched bibliography, the user analyzes the extended metadata and the fulltext of single papers and can decide to remove some papers because not particularly significant or not up-to-date with respect to the research topic. VisualBib enables users to remove single papers through the *bin icon* associated to them (present for example in Figure 2.9, to bottom-right). After confirmation the system will remove it and the existing citation relationships, detaching it from the lists of papers of each involved authors, and deleting the author(s) without associated papers.

2.6.5 Refinement: find new significant papers

Another refinement of the bibliography consists in finding and importing new or up-to-date papers. This can be achieved, for example, exploring the production of an author already present in the diagram to check the presence of updated versions of a known paper: this operation is performed in VisualBib by clicking on an author name and subsequently on the search icon and on the appropriate source icon, as shown in Figure 2.13. The author's productions will appear in a list where the papers, already present in the

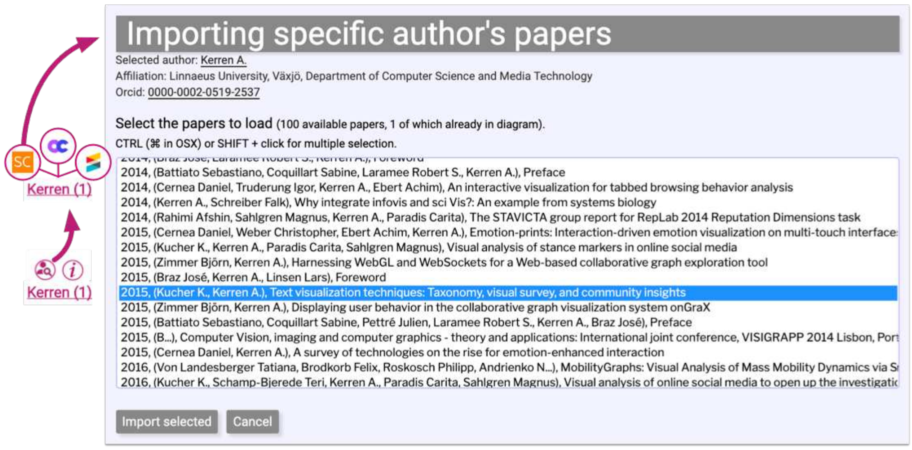

Figure 2.13: Exploring the production of an author. The papers already loaded in the bibliography are marked in blue.

bibliography, are marked in blue. In our example only one paper is already present in the bibliography. Users can import a set of papers, selecting them.
Another possible approach consists in exploring the cited/citing references of a specific paper by clicking on the four-arrow icon in order to find inspiring papers of new works derived form the considered one. Figure 2.14 shows this situation. Users click on the unique paper published in 2014, then on the 4-arrows icon and visualize the list of 28 papers cited by this paper, and 78 citing it. In Figure 2.14 we selected for importing the papers marked in blue. In both these situations, VisualBib suggests the retrieval of extended metadata for the selected papers.

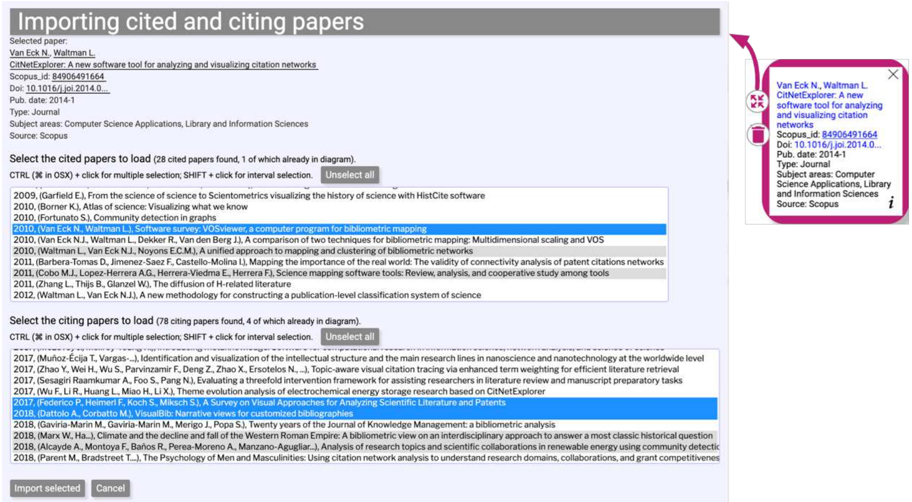

Figure 2.14: Exploring cited/citing references. The items marked in gray are the papers already loaded in the bibliography, the items marked in blue are those selected by the user for importing.

2.6.6 Exporting

When users decide that the bibliography is complete, he/she could export it in BibTeX format. The exported BibTeX will include all the available metadata about the papers and the *authordata* and *references* fields in order to rebuild, in a next import operation, the author and the citations dimensions. For each author in the BibTeX, we export only some metadata, such as *name*, *last name*, and the *authordata*, discussed in Subsection 2.6.1, if present. The other metadata, shown for Kerrer in Figure 2.11, are not exported but they can easily be retrieved by a call to the *Seek metadata* function. Following our example, we export our bibliography and we obtain a BibTeX file[3], where we can observe the richness of metadata.

2.7 Architecture and Implementation

VisualBib is organized as a single page Web application, based on HTML5, CSS3 and SVG (Scalar Vector Graphics) W3C standard languages and Javascript ES6; it makes use of AJAX techniques to perform HTTP/CORS [15] calls to data providers in order to retrieve needed papers and authors metadata. Although the most of the VisualBib Web application runs on the user's browser, a server-side is provided to offer some cloud services as the saving and retrieving of bibliographies, their indexing and sharing besides the tracing of the errors/exceptions in the application.
The timeline of Figure 2.15 shows the advances in the development of the VisualBib platform. An interactive version of the timeline is available at `https://visualbib.uniud.it/en/development/`; all the features introduced in the version 3.0 of the application are presented in the Section 2.9. Figure 2.16 shows the architecture of VisualBib and its main modules, which we describe in next Subsections 2.7.1-2.7.9.

[3]The file generated is available online at http://visualbib.uniud.it/biblio- example-out.bib

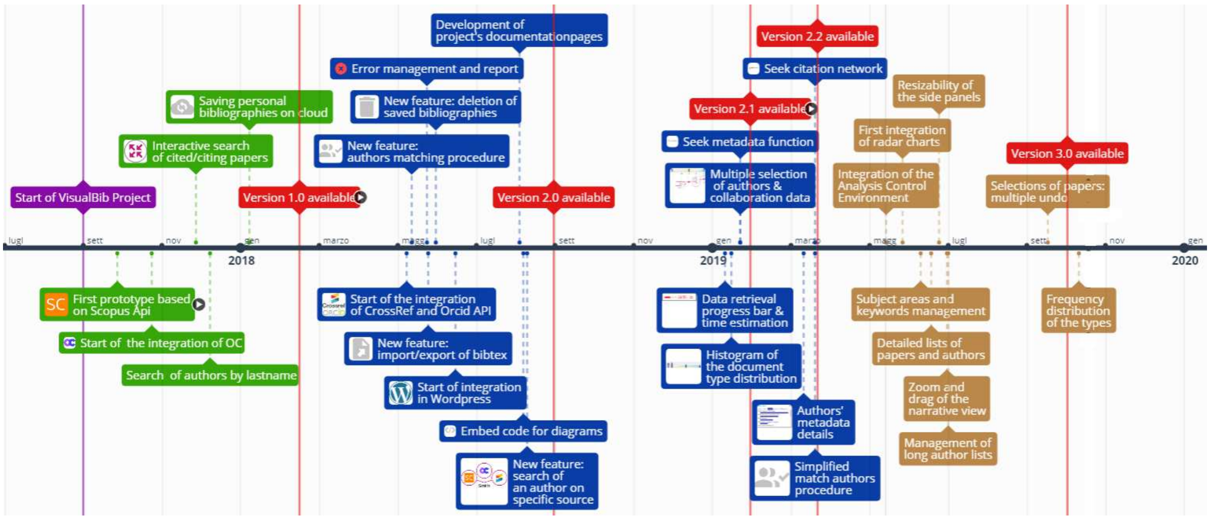

Figure 2.15: The workflow of the development of VisualBib application.

2.7.1 Data providers

In the following we describe specific features and issues related to each provider.

Scopus Scopus marks all publications with a unique record id, called Scopus id that is a numeric code which is part of the EID (Electronic id). Authors are marked with a numeric code called author id. Where available Scopus provides (or can be queried by) universal identifiers for papers (DOI) and authors (ORCiD).

VisualBib interfaces with Scopus through a series of API calls [13] to get needed data:

- *Abstract Retrieval API* to retrieve detailed metadata of a paper given its Scopus id or its DOI;
- *Scopus Search API* to retrieve the references to cited and citing papers of a specific publication;
- *Author Retrieval API* to get detailed metadata of an author and a list of his/her publications;
- *Author Search API* to search authors given their name and surname.

Being the calls generated on the client side of the application, Scopus only accepts API requests originating from domains of subscribers, according to the data policy Furthermore, to avoid misuse of data, the platform introduces some limitations to each API endpoint by fixing quotas for the number of queries per week and per second.

OpenCitations VisualBib implements the retrieve operations from the OpenCitations Corpus [20] submitting queries to a single API endpoint which accepts SPARQL queries. VisualBib uses 8 different query types to get all the needed metadata aabout papers, citations, authors searched by name or ORCiD and their lists of publications. A known issue in the OCC indexes is the presence of multiple identifiers for a same author: in this case it is necessary to consider all the candidates, examine and import related publications and finally join the duplicate author entries by applying the "Match author" procedure.

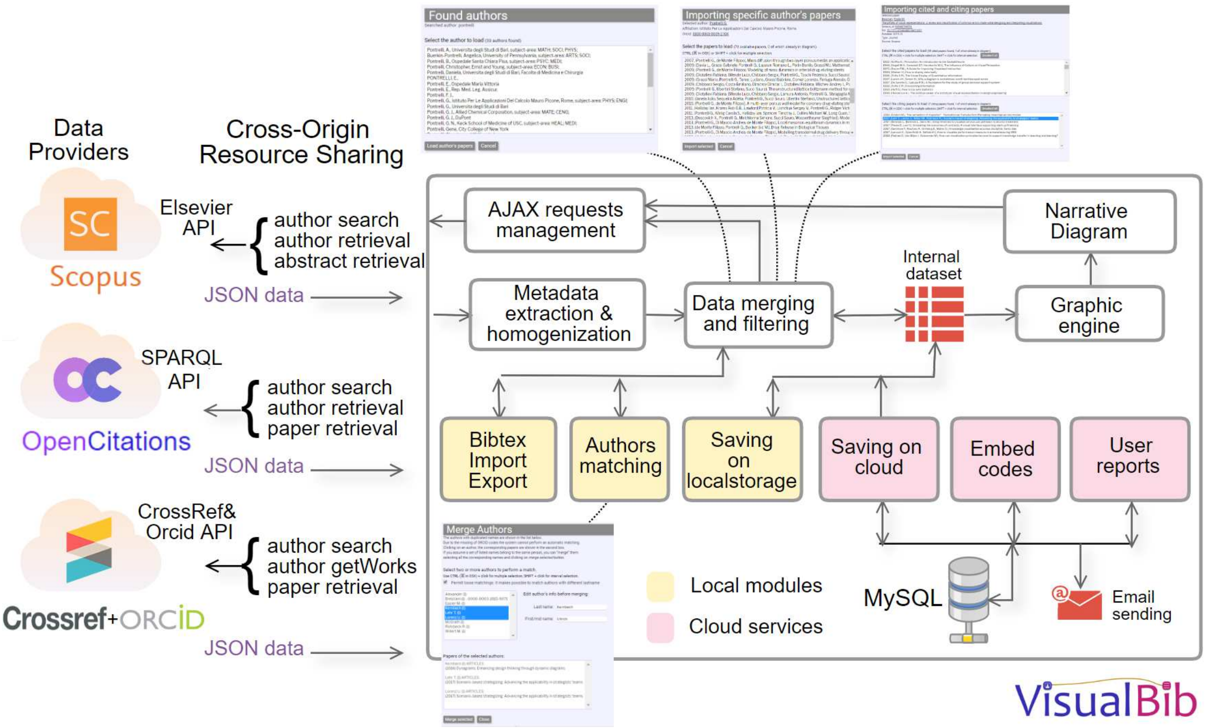

Figure 2.16: The architecture of the VisualBib application.

CrossRef and Orcid Observing the complementarity of the APIs provided by CrossRef and Orcid APIs, we combined their services to implement a third data source. In particular VisualBib performs the search of authors by name using Orcid API [22] and then, after the disambiguation of the results, queries both Orcid and CrossRef APIs to retrieve the list of the papers' DOIs of the selected author. The results from the two sources are merged and their metadata are retrieved, one by one, from CrossRef, querying it by DOI. Due to this, the retrieval of the publications of an author from CrossRef/Orcid can result slow because it requires a http request/response for each paper: a stop button is available to interrupt the current operation and obtain a partial list of results. About the search of cited/citing papers of a selected one, currently CrossRef can only provide, if available, the list of its cited papers and not those citing it. Consequently the corresponding list in the selection form, see Figure 2.2, will always result empty.

2.7.2 AJAX requests management

The interaction with external sources to get real-time data implies the execution of cross-site HTTP calls that must be managed through CORS [15] specific headers to overcome the browser security restrictions. VisualBib must receive the *Access-Control-Allow-Origin* header in order to access the data in each response. Being OpenCitations, CrossRef and Orcid APIs open services, the header above is automatically generated by servers without the need of authentication. Scopus APIs need to receive a previously registered API key in each request in order to authorize the specific domain, declared by the client through the *Origin* header that must match the registered one. This module is also responsible for the preparation of the AJAX requests for the various data provider, each characterized by specific endpoints, parameters and request formats. In addition to single calls of APIs it manages the retrieval of long lists of results, generally returned in small chunks, through

missing data in p	**conditions to test**	**actions** pf: paper found	**notes**
publication year	-	p discarded	publication year is required
list of authors	-	p discarded	authors' names are required
internal id and DOI	b.contains(p.title)	T: pf.update(p) F: b.add(p)	title found, paper updated paper imported
DOI	cond1: b.contains(p.id) cond2: b.contains(p.title)	T: no action F: eval(cond2) T: pf.update(p) F: b.add(p)	paper already in biblio paper already in biblio; paper updated paper imported
-	cond1: b.contains(p.doi) cond2: b.contains(p.id) cond3: b.contains(p.title)	T: pf.update(p) F: eval(cond2) T: pf.update(p) F: eval(cond3) T: pf.update(p) F: b.add(p)	paper already in biblio updated paper already in biblio; paper updated paper already in biblio; paper updated paper imported

Table 2.5: Strategy for importing a paper *p* into the bibliography *b*. T=true; F=false.

a series of repeated requests to be triggered recursively. In fact, the asynchronous nature of AJAX calls imposes to manage responses by a rather complicated nested hierarchy of callbacks; in order simplify the structure and avoid the so-called phenomenon of *callback hell* [70] we are planning to restructure the code by means of Promise objects [10].

2.7.3 Metadata extraction and homogenization

This module provides the necessary data extraction from the JSON flow of data coming from the various data providers. The reception is managed asynchronously through a set of callback functions. The loading of long data streams is accomplished by multiple recursive requests managed by the module described above. In order to avoid simultaneous API calls and achieve a clearer status reporting to users, this module block new search requests until the current data receiving and extraction is complete. Furthermore this module is responsible of the homogenization of the data from different sources, for example by decoding the identifiers of the typologies.

2.7.4 Data merging and filtering

During the loading of new authors and papers, users are asked to filter the retrieved data by means of selection lists in order to pick the significant items from the results.

This module performs a series of checks for comparing the new data with the current dataset in order to correctly insert and integrate the new information, possibly avoiding the introduction of redundancy. It also manages the exceptions that could arise and ensures the consistency of the dataset during the importing and the deletions of papers, the merging of duplicated authors and the import from external BibTeX files.

Table 2.5 describes in detail the process of importing a new paper into the current bibliography, in the case of lack of important metadata. The first column specifies the possible missing field(s); the second column specifies, in OOP (Object Oriented Programming) notation, the sequence of tests to perform: each test returns a boolean value (true or false) and, if the test is successful, the found paper is saved in the variable `pf`. The third column specify the actions to be taken in consequence of each test, in OOP notation.

Table 2.6 describes the conditions to test and the actions to accomplish in order to

missing data in a_i	**conditions to test**	**actions***	**notes**
internal author id and ORCiD	cond1: b.contains(p) **	T: eval(cond2) F: b.addAuthor(a_i)	new author entry ***
	cond2: pf.contains(a_i.name)	T: p.connect(a_i) F: pf.addAuthor(a_i) b.addAuthor(a_i)	author locally identified new author for p ***
ORCiD	b.contains(a_i.id)	T: p.connect(af) F: b.addAuthor(a_i)	author identified new author entry ***
-	cond1: b.contains(a_i.orcid)	T: p.connect(af) F: eval(cond2)	author identified
	cond2: b.contains(p.id)	T: p.connect(af) F: b.addAuthor(a_i)	author identified new author entry ***

∗ pf: paper found - af: author found; ∗∗ search by paper metadata (DOI/id/title)
∗ ∗ ∗ possible duplicated author; matching authors wizard recommended

Table 2.6: Strategy for the attribution of each author a_i of the imported paper p into the bibliography b.

add or map each author of an imported paper, and to connect him/her to a new paper. This strategy is applied to each author of a new imported paper, in all the following cases: after a DOI search, in a search by author (after the selection of the papers to import), in a search of the cited and citing papers of a specific paper (after the selection of the papers to import). It is worth noting how the authors' metadata provided in all the above cases can be different, also within the same data source.

2.7.5 Internal dataset

The internal dataset contains all the data needed to represent the current bibliography. The dataset is organized through a set of object oriented Javascript data structures that reflect the zz-structure described in previous Section 2.4. At this lower abstraction level, zz-cells are objects, organized in arrays and connected each others through references.

For papers, VisualBib manages a set of significant metadata: title (with external link), publication year, abstract, authors (with links), subject areas, Scopus or OC ids (with links), DOI (as link), ISSN, references list and source list. The links connect the metadata to the corresponding resource on the external index.

For authors, the dataset contains: first, middle and surname (with link), preferred name, affiliation, ORCiD, local id, subject areas and the list of the considered papers.

2.7.6 Graphic engine and Narrative diagram

To implement VisualBib we have chosen D3.js [38, 32], a modular JavaScript library for creating interactive documents with a strong emphasis on the HTML, SVG, and CSS Web standards. Similarly to JQuery, D3 provides a powerful DOM selection mechanism based on declarative CSS patterns, a rich library of methods to create complex graphical representations and layouts and to act, with the same syntax, both on single DOM elements and on sets of them.

The idea behind D3 is to strictly tie data to HTML or SVG elements, realizing a so-called *data-driven* approach to DOM manipulation. A set of powerful methods, directly applicable to dynamic selections of DOM objects, makes possible to create, update and remove graphic elements in a Web page when data change. Transitions and animations can be defined using specific functions that smoothly interpolate, over the time, the style properties of a set of elements. D3 provides also many helper modules to define dynamic axes and scales, to parse data files, to interact with asynchronous http requests, to manage and transform data and to support rich visualizations through specialized layouts. In order to generate the narrative diagrams, generally characterized by a limited number of elements, we adopted SVG standard which provides all the needed geometric elements to represent the cells and their interconnections.

The graphic engine maps the internal data model into the narrative view; previously informally shown in the previous Figure 2.1, Figure 2.2, and Figure 2.3, and formally in the Definition 1.3.3; the topological constraints, applied by these modules, have been discussed in previous Subsection 2.5.3.

The papers items are represented by groups of SVG elements being rendered by round-cornered squares combined with appropriate icons. Each paper has associated a series of event handlers to manage users actions and apply style properties to the current element and to the connected ones, to make visible paper details and icons to trigger further actions. Transition and animation effects have been introduced to improve the user experience through a progressive highlighting of the semantically connected items. The *authorship* and *citation* relations between papers are rendered with a SVG path element which describes *smooth cubic Bezier curves* connecting the respective icons. The paths, as shown in Figure 2.3, are opportunely stylized and colored and, in order to reduce the complexity of the representation, any multiple relationships (for example same authors for subsequent papers) generates overlapped paths.

2.7.7 Computational load estimation

We estimated the computational load for the four phases of the process of importing a set of P_i papers of a given author into a bibliography of P papers and A authors. Let be:
A_c = number of candidate authors, provided by an author search
P_a = number of papers of an given author, applying a search
then the upper limits on the number of operations are:

1. Retrieve papers and authors data ⇒ # of API calls: $O(A_c) + O(P_a)$
2. Data merging & filtering ⇒ # of comparisons: $O(P_i \cdot P) + O(P_i \cdot A)$
3. Narrative chart generation ⇒ # of comparisons: $O((P + P_i) \cdot A)$
4. Computation of histogram, area chart and authors collaboration counters ⇒ # of comparisons: $O((P + P_i) \cdot A)$

Considering a bibliography of $P = 100$ papers and $A = 100$ authors and $P_i = 50$, the total execution time, for all the three operations 2, 3 and 4, results less than 400ms on an i5-8250@1.6GHz processor; the duration of operation 1 depends on many factors (such as server and network load); but the user can reduce the response providing, i.e., in the author search, both name and surname or a unique identifier. The values considered in this example are compatible with the intended use cases of the application: the interactive creation, refinement and visual analysis of small bibliographies.

2.7.8 Local modules

VisualBib also has a simple mechanism, activable by clicking on the `Save on local` button (see Figure 2.1) to save a bibliography into the localstorage of the browser: it is a permanent (preserved in different sessions), erasable but nor shareable space, useful for frequent savings. The `Author matching` and the `Import/Export in BibTeX format` wizard, whose functions have been introduced in Section 2.2.

2.7.9 Cloud services

In order to store and retrieve the visual bibliographies, VisualBib includes some server side modules, equipped with a MySQL database server and a PHP interpreter. User diagrams are described by title, content represented in JSON format, email address related to the owner, last saving date and two unique urls: every time a new bibliography is saved clicking on the `Save on cloud` button (see Figure 2.1), the user is asked to specify an e-mail address; the system saves the bibliography in a MySql database and generates a couple of unique urls for future accesses and/or for sharing with other users in both read-write and read-only mode. At the same time a HTML embed code is generated to allow users to incorporate the narrative diagram into a personal Web page. Users may also require, clicking on the `Email my bibliog.` button (see Figure 2.1), to receive the list of their saved bibliographies.

2.8 User evaluation

We have carried out two different studies in order to evaluate the impact of our approach.

In both studies, in addition to VisualBib, we have considered the Scopus Web platform [23] for bibliographic searches in order to evaluate some usability aspects of our visual approach compared to a traditional one. The choice of Scopus as the second platform on which to conduct the tests is motivated by the advanced search features that it makes available and the possibility to evaluate both platforms with a common dataset. The first study, described in [46, 55], was a between-subject qualitative study in which the participants were divided in two groups in order to evaluate separately some usability aspects of VisualBib and Scopus Web platforms. The second study involved a larger group of participants who faced with a series of bibliographic analysis tasks on both platforms: we also collected quantitative data about the execution times of each task as well as some feedback on their user experience. In this case, in order to collect more aware opinions and ratings, we chose a within-subject approach involving each participant in the evaluation of both platforms to inform them about the distinguishing features and let them experiment the two alternative approaches before evaluating each one.

It is important to clarify that the tasks performed by the participants involved only a subset of the features of the two platforms; for this reason, the usability results apply only to the considered aspects. In particular, being the platforms rather different, we concentrated in evaluating common features like the effectiveness in performing simple analysis tasks on bibliographic data and some usability aspects. Both studies were carried out on VisualBib 2.0 version which did not include the histogram, the area chart and the counters of the collaborations of the single authors, introduced in version 2.1.

2.8.1 Study aims

The main questions to try to give answer are:

- Is the VisualBib application effective to deal with some specific bibliographic analysis tasks? We made a comparison with the time employed for the same operations in the Scopus Web platform;

- Is there a significant difference in usability between VisualBib and Scopus, computed by a standard questionnaire such as SUS?

- Is the novel VisualBib interface appreciated by users and considered innovative?

- How important are considered by the users some general features and how much are they perceived present in VisualBib and in Scopus?

2.8.2 Study design and data analysis

Participants The participants were recruited on a voluntary basis among undergraduate students, students of the last year of high school participating to university orientation programs, and librarians of University of Udine: altogether they were 93, aged between 18 and 56 years with a mean value of 24.8 and standard deviation of 8.6. The average level of experience in the use of search engines for scientific literature on a scale from 1 to 5, self-assessed by the participants, was 2.4 with a standard deviation of 1.1.

Apparatus and procedures For the evaluation of the platforms, the participants were free to use their favourite browser among those compatible with VisualBib. The application currently has some known incompatibilities with Microsoft Explorer, Safari and old versions of Mozilla Firefox (version<50). VisualBib detectes the browser in use and informs user in case of incompatibility. The browsers chosen by the participants during the evaluation were Google Chrome (90%), Mozilla Firefox (9%) and Opera (1%). Being the application independent from the operating system, participants were free to use any OS on a personal computer with a screen size of at least 15 inches.

Before starting the experiment we organized a live presentation lasting about 45 minutes in order to illustrate both the Web interfaces of Scopus and VisualBib and to demonstrate some use cases to the participants; then we asked them to perform a series of training activities (the same for both applications), consisting of simple bibliographic analysis tasks, similar to those included in the study described below.

Project and results As mentioned before, the within-subject comparative study included two factors (Scopus and VisualBib platforms) and analyzed 19 variables: the time to perform 5 common analysis tasks (T1, ..., T5) on a bibliography, the perceived usability of the platform measured by a SUS_{-01} standard questionnaire, 6 variables on user experience (U1, ..., U4) and aspects of graphical layout (G1 and G2), and, finally, 7 variables (F1, ..., F7) on specific features of the applications.

The questionnaire was organized in two main sections, the first dedicated to VisualBib and the second to Scopus.

In order to get comparable data, the tasks T1 ... T5 were exactly the same in VisualBib and Scopus, except for the provided input data, while, since we consider the tasks T4 and T5 possibly complex to carry out on the Scopus platform, the participantscould leave the answers empty.

Five quantitative analysis tasks The five tasks regarding specific searches required the filling, on a Web form, of a numerical answer. Before performing each task, users were asked to read and understand the question; then to insert the start time (hour, minutes and seconds), find the solution of the task using VisualBib, insert the answer, and finally report the time at the end of the activity. In order to acquire effective times, the form did not accept wrong or empty answers, forcing the user to find the correct value. The proposed tasks consisted in the following search problems, where the notation A_i indicates a generic author described by name, surname, Scopus Id, affiliation and subject-area.

T1 *Most productive year*: consider the publications of the author A_1. In which year did he/she write the highest number of papers?

T2 *Number of publications in collaboration*: consider again the publications of the author A_1 in a specified time interval. How many of them were written in collaboration with author A_2?

T3 *Self citations of a paper*: consider the paper P_1 of the author A_1, published in a specified year. How many times has it been self-cited in other papers by A_1?

T4 *Textual search in a set of authors*: consider, besides the previous author A_1, the author A_2 and the papers written by them, independently or in collaboration, in a specified time interval: how many of them contain a given word in the title?

T5 *Typology of papers for a set of authors*: consider again the authors A_1 and A_2: how many papers of type *Book* did they write, in collaboration or independently, in a specified time interval?

Since we consider the tasks T4 and T5 possibly complex to carry out on the Scopus platform, the participants could leave the answers empty: in this case we discarded the time measurements related to both platforms. Figure 2.17 shows the distributions of the execution times of the five tasks for the two platforms, discarding the higher outlier data to improve readability of the graphs. The boxes represent the interval between

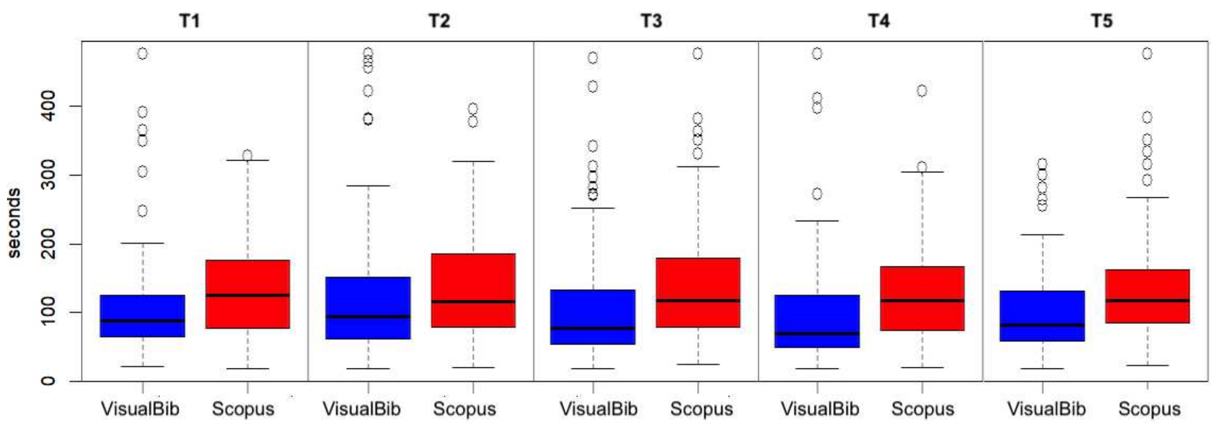

Figure 2.17: The distributions of the execution times of the five tasks for the two platforms.

1^{st} and 3^{rd} quartiles, the black line is the median of the distribution while the circles represents outliers. Having verified that the distributions of the time difference were not normally distributed, we applied the non-parametric Wilcoxon signed-rank test to verify if the difference of execution times on the two platforms were significant for each task. Table 2.7 summarizes the results: considering a 95% confidence interval, we can conclude that the execution times for each task performed on Scopus were significantly higher to those performed on VisualBib platform.

task	sample size*	$\mu(t_{sc} - t_{vb})$	W	p-value	significance level	H_0 hypothesis
1	93	18.3s	2631	0.0002168	0.05	**rejected**
2	93	4.3s	2334	0.03789	0.05	**rejected**
3	93	26.7s	2700	0.002174	0.05	**rejected**
4	83	36.1s	2380	0.0003554	0.05	**rejected**
5	84	52.2s	2290	0.0002627	0.05	**rejected**

* Empty answers on tasks T4 and T5 (optional for Scopus) were discarded

Table 2.7: The results of a Wilcoxon signed-rank test applied to task execution times on VisualBib (t_{vb}) and Scopus (t_{sc}) on the null-hypothesis $H_0 : t_{vb} \geq t_{sc}$

Nine qualitative items related to SUS_{-01} The perceived usability level of the application using a simplified version of the well-known SUS (System Usability Scale) questionnaire [40]. We discarded the first item of the standard SUS, "I think I would like to use this system frequently", to avoid a distortion of the scores in case the system under study is one that would only be used infrequently, and we used the remaining 9 items of the questionnaire with five response options for respondents. Lewis and Sauro [78] studied the effects of dropping an item from the standard SUS questionnaire: specifically, when leaving out the first question, they measured a mean difference from the score the full SUS survey of -0.66 points, considering a 95% confidence interval.

The SUS_{-01} value was computed for each participant and for each platform, with the formula:

$$SUS_{-01} = \left(\sum_{k=0}^{4} (5 - A_{2k+1}) + \sum_{k=1}^{4} (A_{2k} - 1)\right) * \frac{100}{36}$$

where $A_1, A_2, \ldots, A_9$ are the answers to SUS_{-01} items in the scale 1 (strongly disagree)...5 (strongly agree); the odd items refer to negative tone questions, even ones to positive tone questions. The distributions, for the two platforms, are summarized in Figure 2.18. In order to compare the results we applied a hypothesis t-test for the dif-

Metrics/Platforms	VisualBib	Scopus
Min	33.00	0.00
1st Qu.	58.00	33.00
Median	67.00	42.00
Mean	67.8	43.11
Std. dev.	16.04	15.52
3rd Qu.	81.00	53.00
Max	100.00	83.00

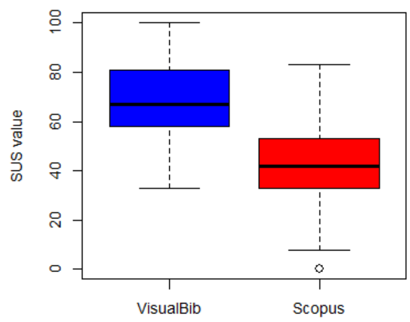

Figure 2.18: The parameters of the SUS_{-01} distributions (left) and their comparison (right).

ference between the means μ_v and μ_s (the VisualBib and Scopus SUS_{-01} means), fixing

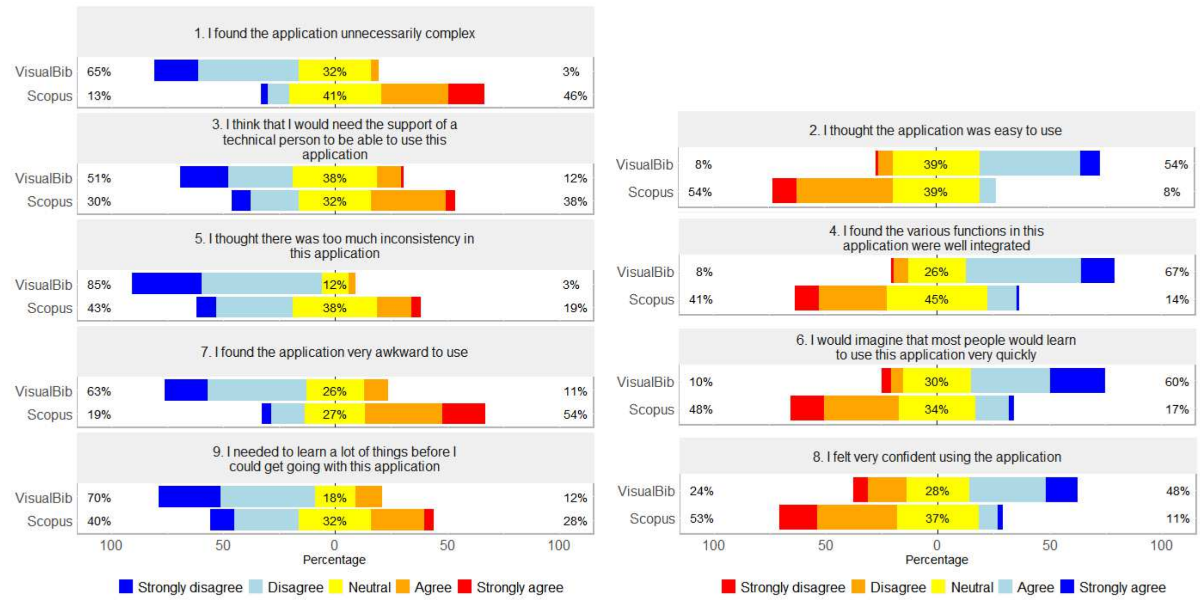

Figure 2.19: SUS_{-01}: the comparative distributions of answers to the odd items, negative tone (left), and to the even items, positive tone (right).

the null hypothesis $H_0 : \mu_s \geq \mu_v$. We have previously verified the normal distribution of the two samples using the Shapiro-Wilk normality test obtaining the W test statistics $W_{scopus} \simeq 0.987$ and $W_{visualbib} \simeq 0.972$ that are within the 99% acceptance interval [0.9631, 1.0000] of the normal distribution hypothesis.

The test statistic $t \simeq 10.8$ corresponding to a p-value$\simeq 10^{-5}$ which is less than the chosen significance level $\alpha = 0.01$ leading us to reject H_0 in favor of the alternative hypothesis $H_1 : \mu_s < \mu_v$. Regarding the absolute values of SUS_{-01} means, their relatively low values probably reflect the difficulty of a part of the participants in dealing with bibliographic search tasks. Figure 2.19 shows the comparative distribution of the answers to single SUS_{-01} odd and even items.

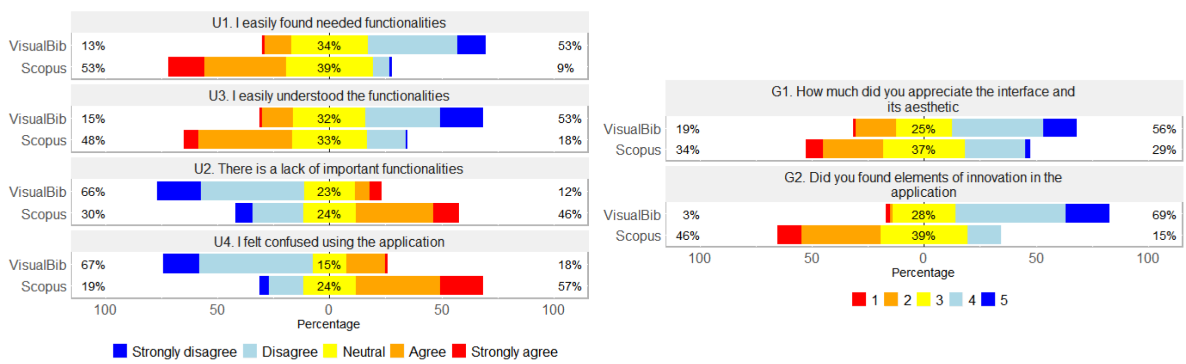

Figure 2.20: The comparative distributions of answers to the U1, ..., U4 (left), and to G1 and G2 (right).

Six specific qualitative items The items are described in Figure 2.20; four of them focused on the user experience (U1, U2, U3, U4 and the last two on the aesthetic and the innovative aspects of the graphical layout (G1, G2). The same Figure 2.20 shows the distributions of the answers for the U1, U3 (positive tone); U2, U4 (negative tone); and G1 and G2.

Seven features The participants were asked to attribute a *value* to seven general features (F1, . . . , F7) and then to quantify the perceived level of each feature's presence in the two platforms. The seven features taken in consideration are present in Figure 2.21,

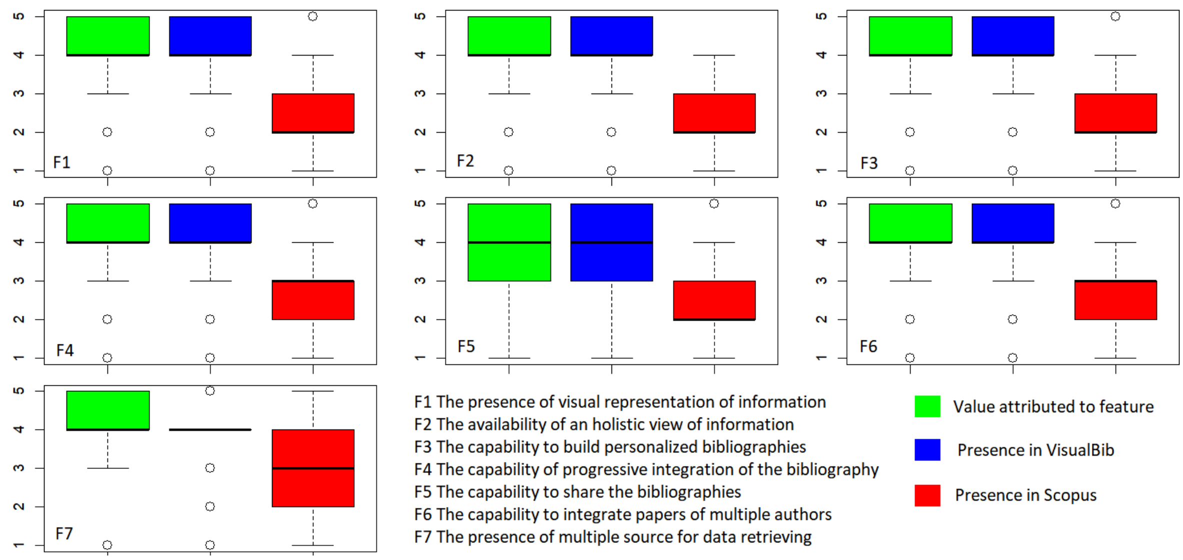

Figure 2.21: The distributions of attributed values to features and of the perceived level of presence in both platforms.

which also summarizes the given answers with regard to the level of importance and of presence attributed by the users to the 7 identified features. For the value attributed to features, the 5 level descriptors come from *1=not important at all* to *5=absolutely important*, while, for the level of presence in each application, from *1=not present at all* to *5=predominant*.

2.9 System advances: VisualBib, version 3.0

In this section, we present the version 3.0 of VisualBib, which introduces new features for the management and the visual analysis of bibliographies:

- support in the quantitative analysis of bibliographic data;
- visualization of detailed metadata for papers and authors;
- a visual analysis tool for the comparison of metrics associated with papers and authors;
- a mechanism for selecting papers according to different criteria, for exploring the co-occurrences of metadata, for analyzing the aggregate data of groups of papers/authors and for the semantic tagging of papers.

As shown in the workflow, outlined in Figure 2.15, the date of publication for this new version has been October 2, 2019.

Next Subsection 2.9.1 presents the new version whose user interface is organized in sections called *environments*; they group and provide easy access to the new analysis tools and full control on the views.

Subsection 2.9.2 presents a user evaluation study to measure the perceived usability of the renewed platform and collect user opinions about new feature and procedures. We will show how all the new features introduced in the last version, while greatly improving the analysis capabilities, do not significantly degrade the level of usability of the platform.

Finally, Subsection 2.9.3 illustrates a data analysis carried out on the zz-bibliography introduced in Section 1.4 using the last version of VisualBib.

2.9.1 The environments

Figure 2.22 presents the new layout of the user interface which is organized in five environments:

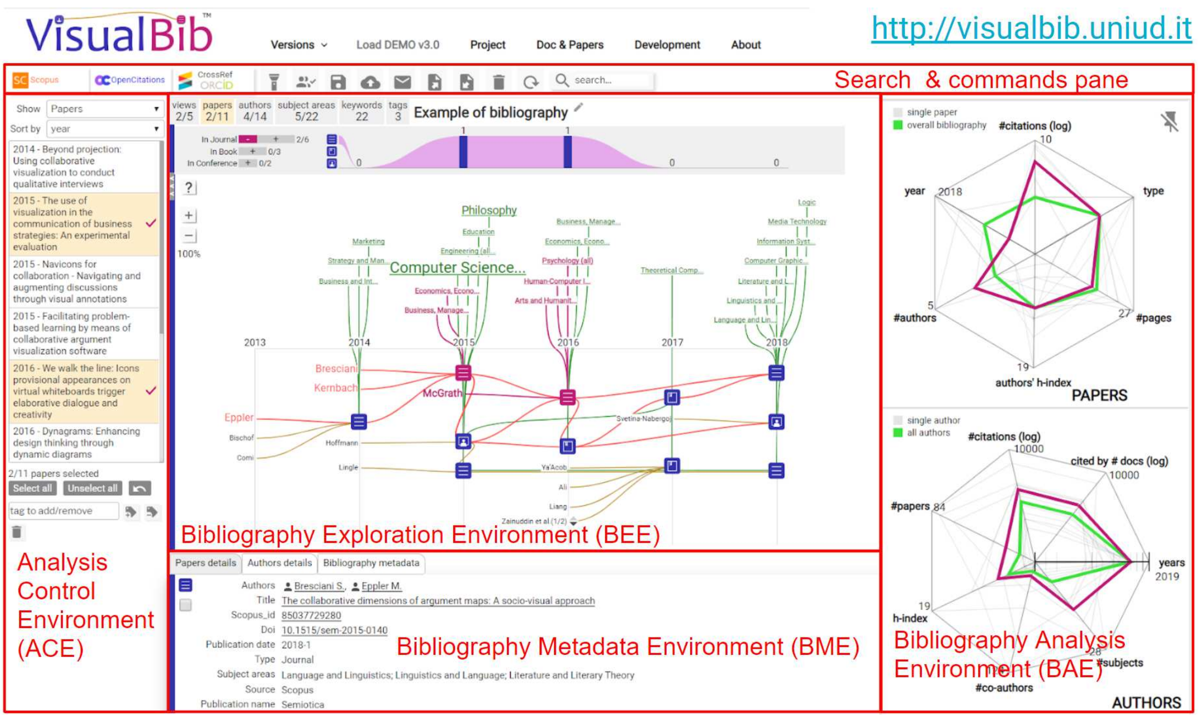

Figure 2.22: The user interface of the version 3.0 of VisualBib. The sections containing the new environments are highlighted in red.

1. The **Search and commands pane** groups a series of buttons both to carry out searches on the supported bibliographic indexes and to execute specific functions like *seek metadata*, *match authors*, *save* and *import/export* operations.

2. The **Analysis Control Environment (ACE)** enables user to explore the lists of the items (papers, authors, subject areas, keywords, tags), to change the sorting criteria, to activate/deactivate the supported views, to display the frequency distributions of the elements, to select/deselect papers and to apply/remove tags to/from papers.

3. The **Bibliography Exploration Environment (BEE)** includes the main narrative view and, on request, the citation network and the wordclouds related to subject areas, keywords and tags.

4. The **Bibliographic Metadata Environment (BME)** collects all available data about authors and papers.

5. The **Bibliographic Analysys Environment (BAE)** shows two radar charts: the first related to papers and the second to authors. Each radar includes a series of metrics, which may be visualized considering single papers/authors, the current selection of papers, or the overall bibliography.

Each environment handles a variety of user interactions, potentially propagating their effects to the others environments, in order to maintain then synchronized; for example, a selection of papers in the ACE leads to visual changes and appropriate highlights in the BEE, BME and BAE environments.
In the following we briefly present the main features of the new environments, starting with the *ACE*: Figure 2.23 shows its various sections, each accessible through the "Show" selection box (on the top-left corner), by clicking on "subject areas", "keywords" or "tags"; or by clicking on the corresponding counter box on the top-left corner of the BEE, shown in Figure 2.22.

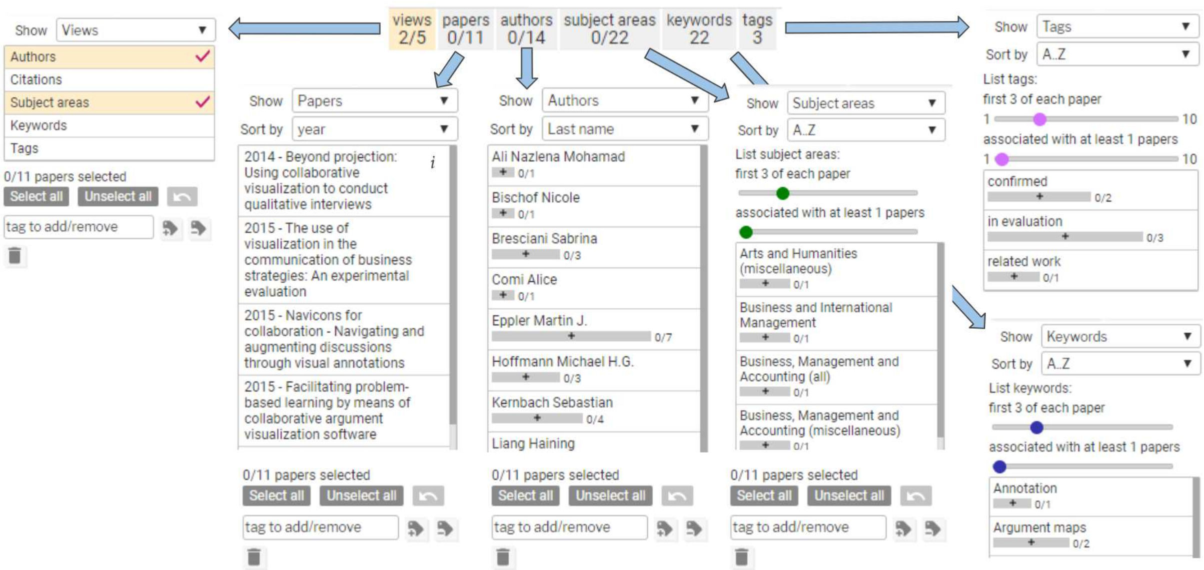

Figure 2.23: The various sections of the ACE environment. From left to right: the panel for the activation/deactivation of the views and the panels with the list of papers, authors, subject areas and tags. The components on the bottom of each section enable users to select/deselect all the papers, to undo the last selections, to apply/remove tags to/from the selected papers and delete them from bibliography.

By selecting the "papers" section (Figure 2.23-second column), a list of all the papers in the bibliography is displayed enabling user to:

- change the sorting criteria (year, title, number of citations, number of authors and type);

- inspect the detailed metadata of a paper on the BME by clicking the “i” icon, visible in the environment ACE by moving the mouse over a specific paper. In Figure 2.23, it appears on the right of the paper ”2014 - Beyond projection: Using...”; in alternative user can click the “i” icon placed in the popup window of each paper in the BEE;

- add each paper to the current selection by clicking on it; alternatively, in the BEE, user can check the selection box included in the detail popup window of each paper.

The others sections visualize respectively: the list of authors, subject areas, keywords (both author and system attributed) and tags.
Each section has its own list of sorting criteria and displays a bar chart representing the absolute frequencies of each element (author, subject area, keyword or tag) in the current bibliography. The bars are marked with a clickable “+” icon in order to let user adding all the papers related to the current item in the current selection. Figure 2.24 shows some details of the ACE panel.

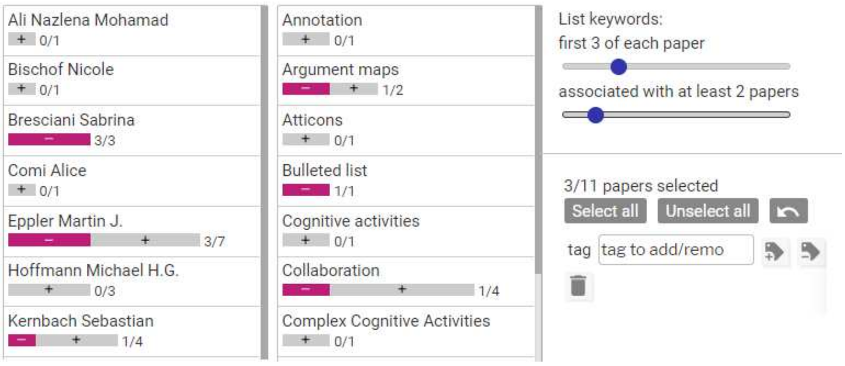

Figure 2.24: A partial view (left) of the author list after the selection of the papers of “Bresciani Sabrina”, in the example. Other authors appear partially selected and the number of papers, written in collaboration with her, are reported. Changing the view to “Keywords” (center), the items associated to the selected papers are highlighted and the number of matching papers is shown. On the top-right we see the sliders for the filtering of the keywords; on bottom-right the buttons to select/unselect all papers, to undo last selection, to apply and remove tags and to delete all the selected papers.

A change in the set of selected papers affects the way in which the bars are displayed: the magenta part of the bars reflects the number of selected papers that are in relation with the current item (author, subject area, keyword or tag). This behaviour enables users to analyze the co-occurrences of items: for example, after selecting all the papers of an author, by clicking on the corresponding “+” icon, the numbers of papers in common with other authors in the bibliography are immediately revealed by the magenta bars.
A “subtraction” operation is also available: by clicking on the “-” icon visible on the magenta section of a bar, it is possible to deselect all the papers related to the current item. In the previous example, we could select the papers of an author excluding and then deselect the ones of a second author in order to highlight the collaborations of the first author with the rest the community excluding the papers co-authored with the second author. The same mechanism is available for subject areas, keywords and tag and can also be used for cross-analysis of different entities: for example selecting the papers of an

author it is possible to analyze the co-occurrences of subject areas, keywords and tags. For these entities a filtering section is also provided in order to restrict the number of keyword visualized in the list below and in the corresponding wordcloud in the BEE. The filter can be regulated by means of two sliders: the first one to limit the extracted subject areas / keywords / tags to the first n attributed to each paper, usually the most significant ones. The second slider permits to restrict the list those items appearing at least in n papers in order to hide the less frequent ones.

Below the lists of items, there is a small panel to apply/remove tags to/from the selected papers. This offers a way to semantically group papers according to different criteria: for example, user can specify a "rule" for some paper in the bibliography (e.g. related work, reference to a theory / method / tool, etc.) and/or to mark a status (in evaluation or confirmed paper, in trash icon, etc.) and or a relevance (highly significant paper or secondary reference, etc.).
The mechanism of selection of group of papers by clicking on authors, subject areas, keywords or tags facilitates the visual analysis all the connections in the BEE are highlighted and the aggregated metrics are displayed in the BAE, as we illustrate below.

The BEE environment reflects the main section of the version 2.2 with some improvements.

- new summary counters; they are visible in the top-left corner and now act as selectors of views; they also show, together with the totals, the number of elements related to the set of selected papers;
- the main view is zoomable and horizontally scrollable;
- any long list of authors is truncated and a new scroll control mechanism is provided;
- the publication year labels are now clickable to easily add the related papers to the current selection;
- the area chart has been completed, on the left, with a clickable histogram which shows the total number of documents for each type. By clicking the "+" and "-" icons on the bars, user can add/subtract corresponding papers to/from the set of selected ones, using the same mechanism adopted in the ACE;
- the ability to display subject areas, keywords or tags wordclouds above the timeline by activating the corresponding views. The font size of each term is proportional to the number of papers related to it; the number of terms in the wordclouds can be adjusted acting on the two filter slider in the ACE. Each term in the wordcloud is connected with all the related papers by a path whose color depends from the active view: green for subject areas, blue for keywords and violet for tags. The single paths can be highlighted, together with the involved papers icons, by moving the mouse over the wordcloud (or over the corresponding item in the ACE).

Figure 2.25 shows a partial view of the BME where users find the detailed metadata sheets of papers and authors in the bibliography. For a direct access to the metadata of a specific author, user can click his/her name in the BEE and then on the "i" icon appearing above it. It is worth noting that the titles and the authors in the BME are links to external pages related to the source of the data, while the author icons, close to the names, allows to open the corresponding metadata page on the second tab of the BME.

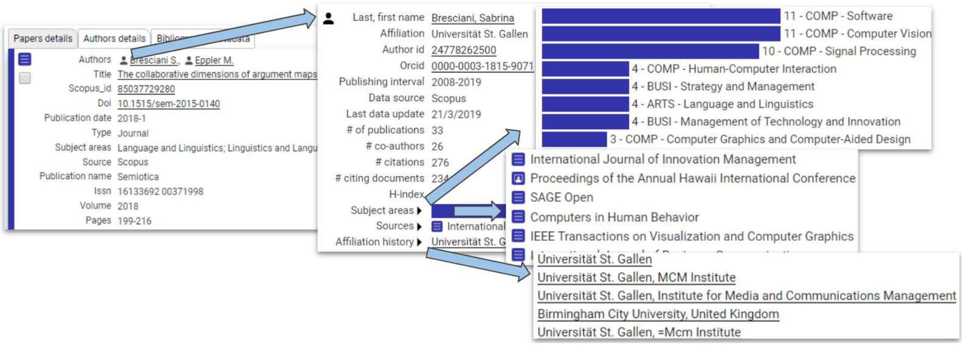

Figure 2.25: A partial view of a metadata detail sheet in the BME related to a paper and an author ("Bresciani Sabrina" in the example). Among the author metadata, the frequency distribution of subject areas, the source where he/she published papers and the affiliation history are visualized.

The amount of metadata available depends on the data provided through a BibTeX archive during an import operation and on those retrieved from bibliographic indexes. In particular Scopus generally provides rich metadata for the indexed papers and authors through the *seek metadata* command.
The *seek metadata* operation can be applied to a single paper and related authors (by clicking the "torch" icon in the popup window or importing a new paper from citations or author search forms), to a group of papers (importing a series of papers of an author or a BibTeX archive) or to the entire bibliography (by clicking the "torch" icon on the command pane). In order to avoid bulk and repeated requests to Scopus API, VisualBib implements a cache mechanism to prevent the updating of papers and authors metadata already retrieved in the last 30 days. Another feature added in the last version is the possibility to annotate the single papers by compiling the "User's notes" field in the paper sheet.
The last environment we present is the Bibliographic Analysys Environment, visible in Figure 2.26, which contains two radar charts, the first displaying six paper metrics and the second one seven author metrics. In particular the papers metrics, associated to the six axes of the radar chart, are:

1. *# of authors* of a paper;
2. *year* of publication of a paper;
3. *# of citations* of a paper, represented on a log_{10} scale;
4. *type* of the paper, with possible values "Conference", "Journal", "Book" or "Other";
5. *# of pages*, namely the length of the paper, if available;
6. *authors' h-index* which is calculated as the median of the h-indexes of all the authors of a paper.

The first radar chart includes a path for each paper in the bibliography (gray lines) and a path representing the median values of the metrics for the overall bibliography (in green). If user selects some papers of the bibliography, the radar presents also the aggregate path (in magenta) based on the medians, for each metric, of the selected papers metadata.

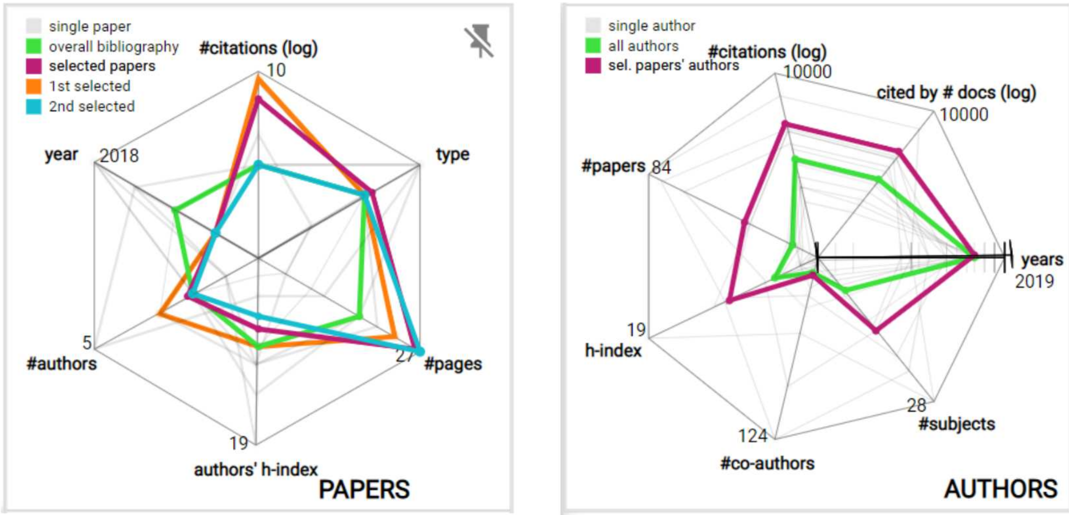

Figure 2.26: The radar chart related to the papers (left) and to the authors (right) of the bibliography.

If the number of selected papers is exactly 2, for the purpose of facilitating their comparison, the radar displays also two additional paths related to the selected papers, highlighted respectively in orange and azure colors. In order to highlight the path of a specific paper and visualize the values of its metrics, user can move the mouse over the corresponding item in the ACE or over the its paper icon in the BEE. It is also possible to interact with the legend or directly with the chart, moving the mouse over the single paths: in this case the path are highlighted in black and the metric values are displayed. This features enable users to easily point out papers with particular values for a metric: the title of the paper appears in the legend and the corresponding icon in the BEE is highlighted.
The BAE panel can also be resized for a better resolution by dragging the handle provided or it can be unpinned to easily move it on the webpage.
The second radar includes the following author related metrics:

1. *# of papers* i.e. the total number of publications of an author, not limited to the papers in the current bibliography;

2. *# of citations* of an author, on a log_{10} scale, related to all his/her publications (not limited to the papers in the current bibliography);

3. *cited by # docs* i.e. the number of documents that cite an author, on a log_{10} scale (not limited to the papers in the current bibliography). This value is always lower than the previous one because a document can cite more than one paper of a same author;

4. *years* i.e. the interval of years in which an author has published; both the median value and a representation of the interval are shown;

5. *# subjects* i.e. the number of different subject areas associated to all the papers of an author. A detailed list complete of frequency distribution is visible in the BME;

6. *#co-authors* i.e. the total number of collaborations in the scientific production of an author;

7. *h-index* of an author.

The radar charts report the data retrieved from Scopus by the *seek metadata* function or those provided by users in a BibTeX archive. Possible errors in the data retrieved from Scopus can be corrected by exporting the enriched version of the bibliography, then modifying the BibTeX archive and finally re-importing it.

2.9.2 User evaluation

In order to evaluate the new 3.0 version of the VisualBib platform, we carried out a user study with the aim of trying to answer to the following questions:

1. How difficult is for users without a specific training on the platform to carry out some analysis tasks on a bibliography ?

2. Is there a significant difference in usability between the 2.0 and 3.0 version, computed by a standard questionnaire such as SUS?

3. How effective users find the various sections of the application in performing simple analysis tasks?

Participants We recruited 25 participants on a voluntary basis among undergraduate students, professors and librarians of University of Udine, 16 males and 9 females, aged between 20 and 57 years with a mean value of 29.5 and standard deviation of 11.8. The average level of experience in bibliographic research of scientific literature, self-assessed by the participants on a scale from 1 to 5, was 2.4 with a standard deviation of 1.0.

Apparatus and procedures For the evaluation the participants were free to use their favourite browser among those compatible with VisualBib. The browsers chosen by the participants during the evaluation were Google Chrome (84%), Mozilla Firefox (12%) and Microsoft Edge (4%). Being the application independent from the operating system, participants were free to use any OS on a personal computer with a screen size of at least 15 inches.

Before starting the experiment we organized a live presentation to illustrate the various sections of the Web interface and to show and test a use case scenario consisting in:

- **importing** a BibTeX archive containing 18 bibliographic references with limited metadata (see Appendix A.1);

- **enriching** the metadata of the bibliography, by using the 'seek metadata' feature, and analysing the results and possible issues with duplicate or misspelled author names;

- **matching** the author names, using the'match authors' feature, in order to detect and merge duplicated author entries and correct misspelled names;

- **integrating** the bibliography with additional papers looking through the scientific production of one of the authors or exploring and importing cited and citing papers of one of the paper in the bibliography;

- **exporting** the enriched bibliography in BibTeX format: in the output archive, listed in the Appendix A.2, all the available metadata about papers are reported in the appropriate fields, together with further non-standard fields necessary to reconstruct, in VisualBib, the correct authors/papers relationships and the citation network;
- **analyzing** the final bibliography, extracting some metrics such as the number of papers, the number of papers with specified subject area or keywords, their co-occurrences, the number of citations of the authors and their h-index, the most productive author, the most cited paper, the distributions of paper types, etc.

After the training activities we provided participants with a shared bibliography[4] on which to carry out a series of analysis tasks and ask them to fill an anonymous questionnaire.

Project and results We planned the questionnaire in order to measure the difficulty in carrying out some analysis tasks on a bibliography using VisualBib 3.0 and to collect data about the usability and the effectiveness of the procedures offered by the platform. It was organized in four sections:

1. collection of personal data (gender, age, level of experience in bibliographic research) and about the browser used in the test;
2. proposal of 8 tasks for the analysis of the bibliography and the collection of the related answers;
3. collection of the answers to the 9 questions of the simplified SUS_{-01} test;
4. collection of users opinions on the effectiveness of the various features and procedures of the platform like importing, enriching, normalizing, integrating and analyzing a bibliography.

The set of tasks relating to point 2, consisted in finding the answers to the following questions:

T1 the number of authors in the bibliography;

T2 the maximum number of authors per paper;

T3 the most frequent subject area of the papers;

T4 the most frequent keyword among the first attributed to papers;

T5 the author with the highest impact index (h-index) among those in the bibliography;

T6 the average h-index of the authors of a specific paper;

T7 the number of authors of conference papers;

T8 the number of papers associated with 2 specific subject areas.

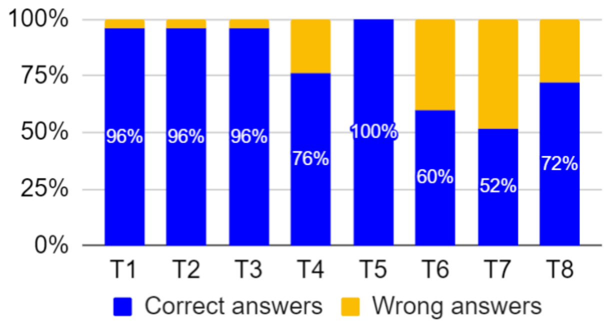

Figure 2.27: The rate of the correct answers to the tasks T1, ..., T8.

Figure 2.27 shows the percentage rate of correct answers to the tasks T1, ..., T8. The tasks T1, T2, T3 and T5 resulted easy to carry out (correct answers≥96%); the T4 task involved the use of one of the filtering sliders in the ACE keyword section and scored 76% of correct answers. The remaining tasks implied the finding of second-level metrics that required 2 or more operations to be performed: T6 needed to read one of the radar chart metrics after selecting the right paper (60% of correct answers), T7 involved the selection of specific paper type by clicking on the related bar on the left of the top area chart and then read the author counters (52% of correct answers) and finally, T8 implied the selection of a subject area and the reading of co-occurrences with a second subject area on the bar chart. The rates of correct answers reveal the difficulties for user to carry out non-trivial tasks without a specific training: users participating to the study had less than an hour to experiment the platform before try the test. Figure 2.28 illustrates the distributions of the SUS_{-01} for the two version of the platform. We applied a hypothesis t-test for the difference between the means $\mu_{v2.0}$ and $\mu_{v3.0}$ (the SUS_{-01} means of the VisualBib 2.0 and 3.0 versions). The test statistic $t \simeq 0.73$ corresponds to a p-value$\simeq .23$

Metrics/Platforms	VisualBib 2.0	VisualBib 3.0
Sample size	93	25
Min	33.0	28.0
1st Qu.	58.0	58.0
Median	67.0	64.0
Mean	67.8	65.0
Std. dev.	16.0	19.1
3rd Qu.	81.0	78.0
Max	100.0	100.0

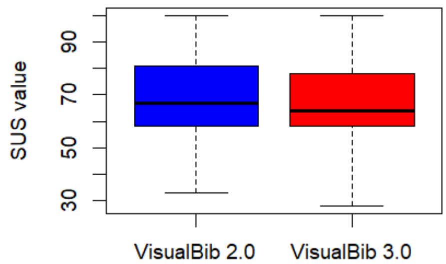

Figure 2.28: VisualBib 2.0 and 3.0 SUS_{-01} test results: the parameters of the distributions (left) and their boxplot comparison (right).

which is greater than the chosen significance level $\alpha = 0.01$ leading us to state the not significance of the difference between $\mu_{v2.0}$ and $\mu_{v3.0}$. We can state that integrating all the new features of the VisualBib 3.0 expands the analysis capabilities of the platform without significantly decrease its perceived usability.

Figure 2.29 shows the distribution of the answers of the single SUS_{-01} questions, grouped by positive and negative tone. Most positive results regarded the *Q5: lack of inconsistencies in the application* (92% of the sample agree or strongly agree), *Q4: the well-integration of the various functions* (84% of positive answers). The answer with most negative responses was *Q3: I think that I would need the support of a technical person to*

[4] The bibliography used in the evaluation can be accessed at `http://bit.ly/vb3-evBib`

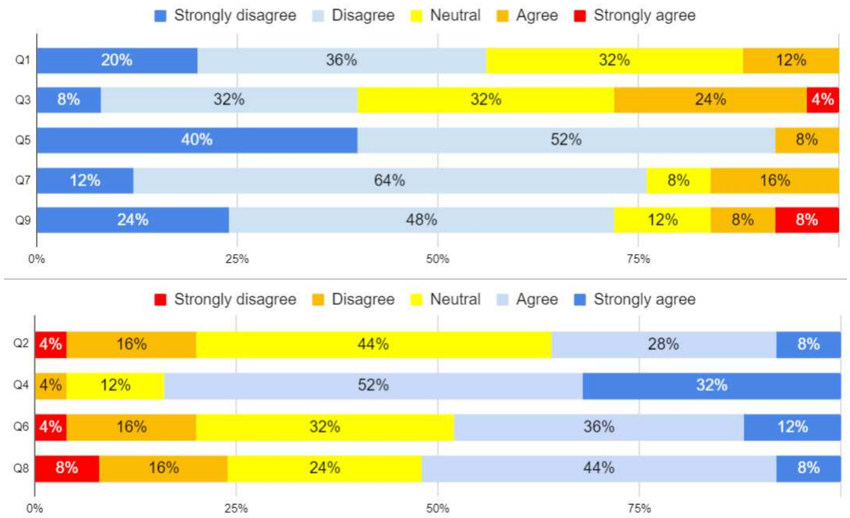

Figure 2.29: The distribution of the answers to the single SUS_{-01} negative-tone questions (top) and positive-tone questions (bottom).

be able to use this application that got only 40% of positive feedback (strongly disagree / disagree responses). This is probably due to the difficulty in approaching bibliographic analysis tasks, a rather uncommon task for part of the participants.

The last part of the questionnaire collected the users opinions, on a 5-levels scale, on the effectiveness of the sections and procedures of the platform and its overall appreciation, in particular about:

A1 the use of narrative diagrams for the representation of bibliographies;

A2 the procedure for importing and enriching a bibliography;

A3 the procedure for resolving authors' duplications in a bibliography (*Match Authors*);

A4 the procedures for integrating the bibliography with further papers by an author and cited/citing papers;

A5 the bibliography Analysis Controller Environment (ACE - left section);

A6 the Bibliography Analysis Environment section (BAE - right section related to graphs) for the comparative visualization of the metrics of papers and authors;

A7 the degree of overall appreciation of the platform.

Figure 2.30 shows the collected data on the 7 questions: all aspects were positively assessed gaining at least 72% of positive scores (levels 4 or 5 of the scale); the most positive evaluation was about the overall platform (84% of positive scores); the least positive evaluation (72% of positive scores) was about the radar section of the application, probably not so easy to interact with for a part of the participants.

2.9.3 Case-study: the zz-structure bibliography

In this subsection, we present a use case scenario consisting in the visual analysis of a bibliography, specifically the collection of references on zz-structures research, presented

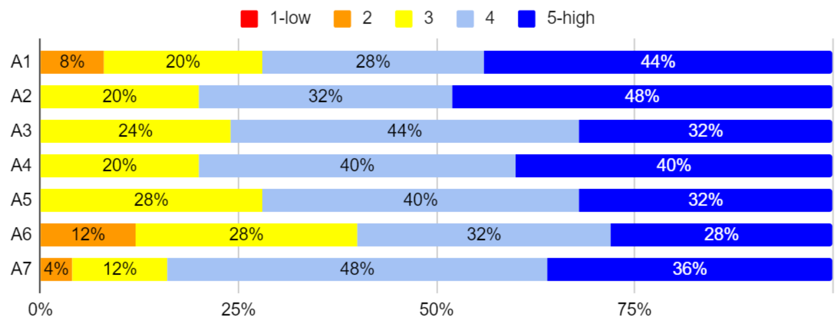

Figure 2.30: The distribution of perceived effectiveness of sections and procedures of the applications. A7 refers to the overall appreciation of the platform.

in Section 1.4. After importing the references in the VisualBib 3.0 platform by means of a BibTeX archive, we provided to enrich it by the *seek metadata* function and to merge duplicates authors names by the *match authors* function.
Figure 2.31 shows the generated narrative-view of the bibliography, limited to the author dimension. Figure 2.32 presents the citation network of the bibliography that is built

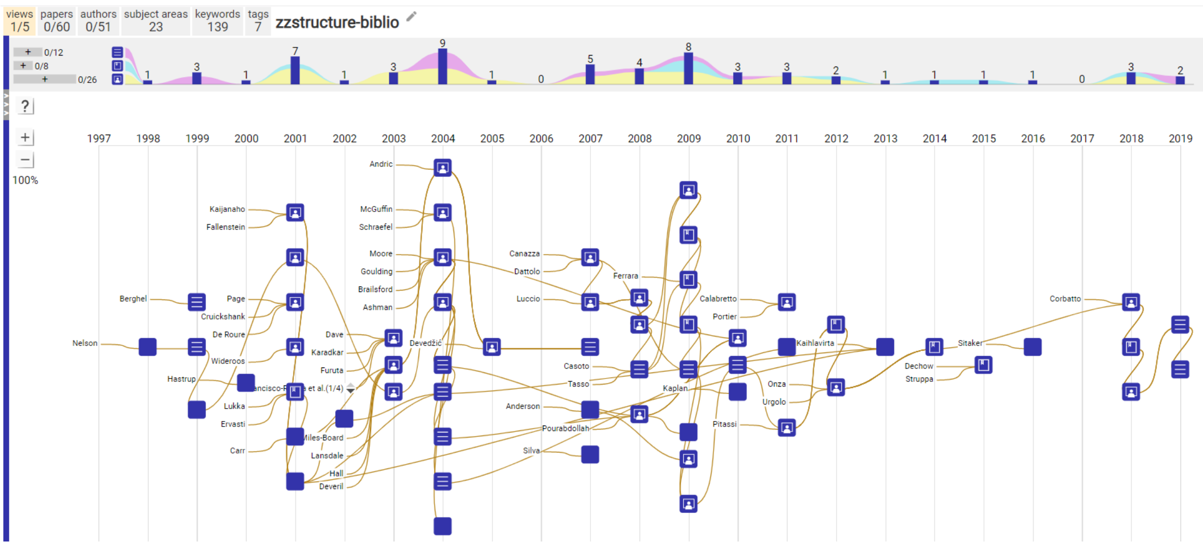

Figure 2.31: The main narrative view of the zz-bibliobgraphy limited to the author dimension.

on the basis of the citation data for each individual paper provided by Scopus. In order to semantically group the papers in the bibliography, we defined some tags and applied them to subsets of the papers: *application* (or *applitude*), *documentation*, *formalization*, *implementation*, *state of art*, *secondary reference*, *vision*.

Figure 2.33 shows a partial view of the wordclouds related to subject area, keywords and tags generated by the system when the corresponding views are activated. In total the bibliography includes 28 subject areas and 233 different keywords but they can be filtered by means of the two sliders described in Subsection 2.9.1 in order to improve the readability of the diagram.

Table 2.8 describes 14 of the possible analysis tasks that can be carried out in VisualBib 3.0. For each of them the result in the context of the considered bibliography is reported together with the necessary user interactions to get the data and the involved

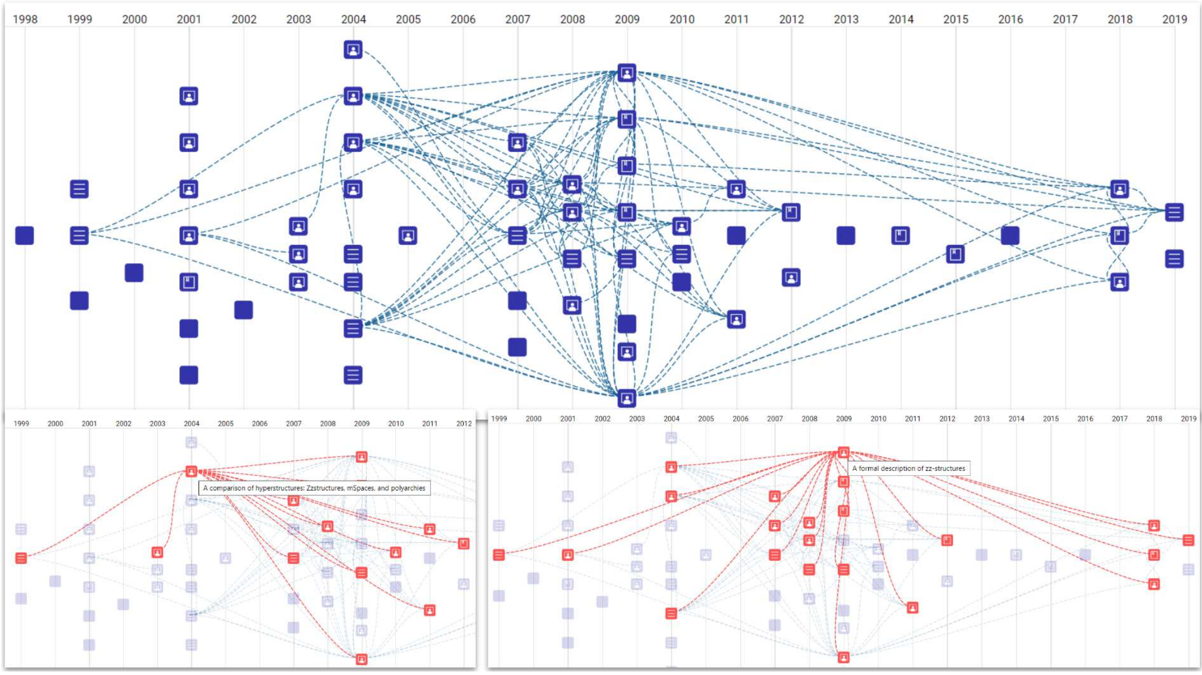

Figure 2.32: An overview of the citation network built on the basis of the citation data for each individual paper provided by Scopus. At the bottom the view of references and citations of the two most frequently cited papers in the bibliography, obtained moving the mouse over them.

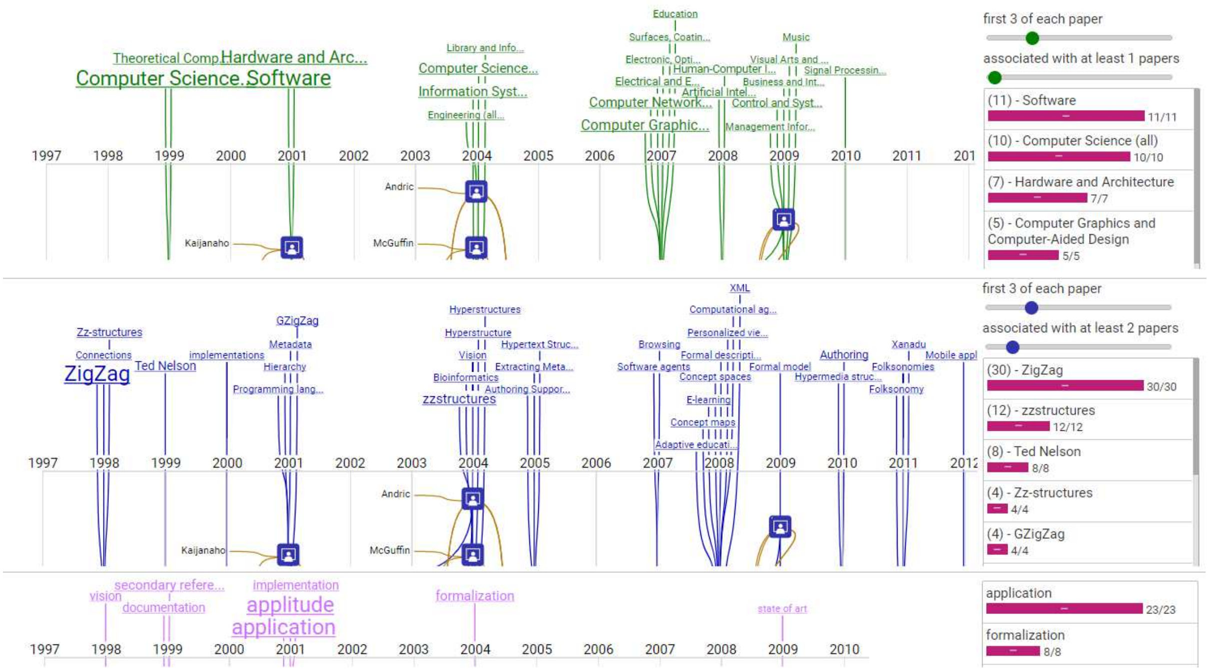

Figure 2.33: A partial view of the filtered wordclouds related to subject area (top), keywords (middle) and tags (bottom); on the right are visible the first part of the lists of the most frequent items as provided by ACE. Long terms result truncated but can be viewed entirely by moving the mouse over them or over the related papers.

environments.

The analysis tasks from 1 to 8 are carried out by interacting in a simple way with a single environment; the tasks from 9 to 12 use a single environment but need multiple user interactions in order to be completed; the last two tasks need multiple operations on multiple environments. It is worth noting how the mechanism for selecting papers through different criteria (paper type, year, author, subject area, keyword or tag) enables users to easily perform cross analyses on heterogeneous keys: for example, selecting one author's papers, the system provides the number of co-authored papers with all the others authors, the distribution of frequencies of all the subject areas, keywords, etc. It is then possible to exclude some papers from the selection on the basis of a second criterion, to add new ones by a further criterion and so on. In this way it is possible to obtain information on the distribution of all the items in the system (authors, paper types, years, subject areas, keywords and tags) on the basis of complex selections. Furthermore the radar charts enable to compare single and aggregate data about papers and authors.

#	Analysis task	Value(s)	Interaction(s)	Environment
1	# of papers and authors	60, 51	check counters values	BEE
2	# of subject areas and keywords	23, 139	check counters values	BEE
3	most productive year	2004, 9 papers	check bars on the stacked area chart	BEE
4	# of papers/type	12 journal, 8 book, 26 conference, 14 unknown/other	check bar chart on top left	BEE
5	most productive year/paper type	2004, journal, conference, 2009, book	1. select a type 2. check bars on the stacked area chart	BEE
6	most frequent tag	application/applitude 23 papers	sort tags by # of occurrences	ACE
7	most cited paper	Xanalogical structure..., 65 citations	sort papers by # of cit.	ACE
8	most represented author	Dattolo, 21 papers	sort authors by # of papers in bib.	ACE
9	most frequent tag in papers published in 2001	implementation, 4 papers	1. click on year 2001 2. display the tags list 3. check longer magenta bar	ACE
10	paper of the most influential group of authors	Linking method for..., median authors' h-ind. 22	1. consider the paper radar 2. select the curve having max authors' h-index	BAE
11	most frequent keyword for the author Nelson	Zigzag, 6	1. select the author, 2. sort keywords by # of occurrences	ACE
12	most influential author about *vision*	Nelson, 4 papers	1. select the *vision* tag 2. display the authors list 3. sort by # of papers in bib.	ACE
13	median of the citations of the papers tagged as *vision*	33	1. select the *vision* tag 2. check the # of cit. of the selected papers in radar	ACE, BAE
14	positioning of the authors of papers tagged *formalization*	higher median values in all metrics compared to all authors as a whole	1. select the *formalization* tag 2. compare red path with green path in the author radar	ACE, BAE

Table 2.8: A list of analysis tasks carried out on the zz-structure bibliography in the VisualBib v.3.0.

2.10 Conclusions and future work

In this chapter, we presented VisualBib, a Web application which offers some original features to support the researchers in creating, saving and sharing their bibliography, starting from a set of papers and authors. After introducing the basic functionalities of the version 2.2 of the application, we analyzed the services offered by a set of bibliographic indexes for real-time retrieval of metadata and proposed a model based on zz-structure for scientific bibliographies and their representation. Then we illustrated a use case scenario to demonstrate how the platform supports some typical user tasks, followed by a description of the system architecture and the various modules that are part of the application. Two user evaluations carried out and presented in this work and in [46] highlights the positive impact of our visual model and the usability of the narrative views. Finally we presented all the new features included in the latest version 3.0 of VisualBib which offers a whole series of tools for the visual analysis of bibliographies. A user study carried out on this version showed how the integration of the new features do not significantly decrease the perceived usability of the application. Finally we presented a case-study which illustrated the potentials of the platform in carrying out a series of analysis task on a sample bibliography.

The VisualBib platform represents an original contribution of this thesis, it introduces novel representations of scientific bibliographies by means of original zz-views and offers users some tools for integrating, enriching, connecting and analyzing the collected metadata.

The limitation of the model mainly concerns its scalability: as the size of the bibliographies increases, the complexity of the views makes VisualBib less effective in exploring metadata and their connections and the system shows some slowdowns in the data visualization and comparison.

Future work will concern the improvement of the system's robustness, the interfacing with additional citation indexes, the management and representation of tagging associated to citations, the integrations of new views and semantic zoom mechanisms and, finally, the improvement of the system evaluation through further studies to collect data and analyze different categories of users (for example librarians, researchers, reviewers, students).

3

AppInventory: a multimedia catalog of resources for active learning approaches

A rapid transformation of methods, roles and practices is currently affecting all school grades. There are many factors contributing to this momentous change: a crisis in traditional teaching methods; the availability of low cost mobile technology and easy access to global knowledge; the strong influence of new technologies on society and communication media and, not least, the desire of educators to find new ways to engage and motivate students.

An increasing number of teachers begins to experiment active learning scenarios and approaches, consistent with proficiency and skills development outcomes as stated, for example, in recent Italian School Reforms and in the European Digital Competence Framework for Citizens [8].

A recent investigation [49], that we carried out involving our target teachers, 178 from high school (K9-K13 grades), middle school (K6-K8), and primary school (K1-K5), highlighted the importance attributed to the design of activities compared to contents design, a marked interest in the role that technology could play in education processes, the importance of diversifying the learning activities and the need for a more extensive knowledge about applications to support the creative work and communication. Above all, the importance of adopting active methodologies emerged.

Nowadays, for implementing their Teaching and Learning Activities (TLAs), teachers have available a huge amount of contents and learning objects on the Web but also hundreds of Web 2.0 and mobile applications, which can support them in creating and sharing digital artifacts, aggregating, remixing and collecting heterogeneous materials and communicating within working groups. These applications represent a challenging opportunity for teachers who would like to experiment and adopt student-centred methodologies and use them into daily TLAs: they can improve the collaborative, cognitive and creative work of the students, enhancing and redefining traditional educational practices. Nevertheless, although these applications are generally easy to find and use, it is often difficult, for a teacher, to find the right one for a specific task, and to have a general awareness on their availability and their potential in an educational setting.

Our work is located in this context and is part of a wider project, called LDInventory [48], which intends to model and realize a novel lightweight Web-based tool for Learning Design (LD). An LD system is a computer based tool which supports teachers in the delicate task of designing, organizing and sharing TLAs with students and colleagues. On such a platform, a teacher can arrange the activities, attach appropriate contents and be guided in choosing relevant tools for the students' tasks. A meaningful module of this project is represented by AppInventory, which this chapter will address.

AppInventory is a digital catalog of (at the moment, 281) Web 2.0 and mobile applications, whose main aim is to support teachers during the design and the implementation of TLAs.

The major *novelty* of AppInventory, respect other existing apps' catalogs, is the graphical modality to visualize the catalog, associated to a semantic mechanism for browsing through it:

- *graphical layout*: the catalog is shown using a unique holistic view, displayed using a multi-resolution circle packing diagram, which starts from a general view of the applications, organized using a taxonomy, and, applying different zooming levels, gets up to the details of the single application;

- *semantic browsing mechanism*: each application is a cognitive unit, semantically connected to the others, by specific, and upgradeable contextual dimensions, such as the complexity and/or Bloom level, the presence of advertising, the typology, the language, and so on. The user can dynamically select the semantic filter to apply on the catalog in order to browse through it following their expectations. We applied semantic structures, called zz-structures [89, 60, 59], and browsing mechanisms based on zz-dimensions and zz-views.

A first prototype of AppInventory has been discussed in [49]; the online AppInventory platform was presented in [47] where we updated the data and created video-tutorials; introduced *statistics* about data and taxonomies; implemented an app's *rating schema* for collecting the opinions of the users, their comments, and suggestions; implemented a new *contextual navigation mechanism* between categories and applications, which is based on zz-structure; performed a qualitative and comparative *evaluation*.

The AppInventory digital catalog of applications represents an original contribution of this thesis both for the underlying semantic model based on zz-structure, which formally introduces a new zz-view for the visualization of the taxonomy, and for solutions adopted for the contextual navigation and interaction with metadata.

The rest of this chapter is organized as such: Section 3.1 discusses related work; Section 3.2 describes our proposal, the cataloging scheme, the purpose-based taxonomy, some statistics on the dataset; Section 3.3 proposes the architecture model and its implementation details; Section 3.4 presents the guidelines followed for the development of the Web platform, the new rating scheme and the semantic browsing; Section 3.5 introduces the zz-structure-based data model of AppInventory and the zz-view for the main user interface of the catalog; Section 3.6 describes some data analysis tools recently developed while Section 3.7 illustrates the interactive guided tour of VisualBib, included in the current online version and modeled through a dedicated zz-structure; Section 3.8 shows the results of two studies to evaluate the usability and the users' opinions on four specific aspects. Conclusions and future work end the chapter.

3.1 Related work

Several repositories exist which index applications, proposing classification [44] and evaluation [45, 76, 68] schemes; in this section, our analysis is restricted to classifications that support teachers in identifying applications for specific purposes, excluding repositories deemed too general, such as App Store, Google Play, Chrome Web Store, Appszoom, or repositories which share learning objects and didactic resources and not tools, such OER Commons.

A positive example is EdShelf [11], a rich discovery engine of websites, mobile apps, desktop programs, and electronic products for teaching and learning. A user can filter the tools by price, platform, subject, age, category and keywords. Unfortunately, subject and category are two long *flat lists* of keywords. Interesting is the opportunity for users to rate and review the apps, and to create and share a shelf of apps. Essediquadro [14] is a service of documentation and orientation on the teaching software and on other resources for the learning process. The tools can be searched by subject of study (Mathematics, Italian, etc.) and by specific subject matter, but *the category of the tools is not considered*. Similar search fields are proposed by Apps4edu [2]. It is possible to list all the apps in it, but the result is *a flat, unusable, paged-list of tools*. CSE (Common sense education) [4] introduces the interesting, abstract concept of *purpose*, but it is used more as teaching context of use more than real purpose.

A comprehensive review of existing application classification systems is provided in [44]; it confirms that a good classification model needs to consider the purpose of the teachers and proposes a classification divided on skill-based, content-based and function-based applications, which implicates respectively the “Remembering and Understanding”, “Applying and Analysing”, and “Evaluating and Creating” levels of the Bloom’s Taxonomy [36]. From our viewpoint, by the term “purpose” we mean the concrete objective of the teacher (or of a generic user), such as realize an infographic, or create a timeline, or plan a quiz. On this basis, we propose, in next Subsection 3.2.3, our original taxonomy.

Related work highlights some open challenges and weaknesses, which represent the start points in the modelling of AppInventory:

- the navigation and searching of tools do not offer a general overview, but long lists of applications, often difficult to read;
- there is a complete lack of graphic views which could offer users a visual, holistic idea of the existing tools;
- the concept of category is often thought of as a subject of study, or context of use and not as purpose for teachers. The existing taxonomies are not purpose-based;
- the semantic relations among the tools are not highlighted, and the degree of belonging of a tool to a cluster in the taxonomies is not clear;
- the interaction with the user, except for EdShelf, is limited to the search box.

Our contribution focused on these objectives and proposes a model and a Web platform which offer graphic and holistic views of the whole catalog, organize the applications in a purpose-based taxonomy, facilitate a semantic navigation among items for the users, and enable users to interact with the platform, rating and reviewing an app, leaving a comment or suggesting a new app.

3.2 Our proposal: AppInventory

AppInventory is an online platform, freely available for research and teaching, not for commercial purposes, at `http://appinventory.uniud.it`. It contains a visual multimedia catalog of 281 applications; it has been developed with the aim of supporting teachers in identifying the best tools to carry out specific tasks, improving the digital skills of teachers and students. In particular, AppInventory has been modelled for:

- providing detailed and multilingual information about each app, including an illustrated review, a video presentation and references to external documentation;
- cataloging the apps by means of an original taxonomy and semantic connections;
- offering intuitive and contextual navigation mechanisms;
- generating visual representations and holistic views of the catalog;
- proposing users some semantic paths through the catalog in order to help users discover new tools;
- inviting users to contribute with evaluation data, reviews, feedbacks, comments and use cases about the presented tools.

The AppInventory project is consistent with the objectives of the European Digital Competences Framework for Citizens 2.1 (DigiComp) [8]: in particular, it can contribute to the development of ten of the twenty-one competence dimensions stated in the DigiComp's conceptual reference model.

3.2.1 Creating the repository

The initial effort has been dedicated to build the App metadata repository, a database containing the multimedia catalog of the applications. The cataloging work has been carried out in two stages: in the first stage, we considered a first set of 111 applications and proposed a classification model; in the second stage we extended the analysis to other 160 new applications.
The selection of widespread and heterogeneous Web 2.0 and mobile applications has been carried out by analyzing several sources, from educational sites and dedicated blogs, such as [16, 12, 11, 67], to thematic link collections and search engines. From the examination of the first group of apps, we have identified common features and purposes of the applications in order to propose an original purpose-based taxonomy and establish a set of features for defining the cataloging scheme.
Subsequently each application has been analyzed and documented through a cooperative work involving a large group of higher education students, 112 for both stages. All the working documents and the coordination sheets have been hosted on a cloud platform, making possible the collaborative editing of documents, their subsequent refinements, the peer reviews of materials and the coordination of the project. A general check was performed from another group of 12 students to assure an homogeneous categorization criteria and the correctness of the collected information. An original video-presentation of each app has been recorded and another group of 5 students looked after their post-production in order to cut the inappropriate parts, add credits, titles, descriptions and tags and publish them on the dedicated play list of the project, accessible from Sasweb Lab's AppInventory project page - `https://goo.gl/25DN6v`, shown in Figure 3.1.

Finally, a group of 8 students contributed to the creation of English subtitles for the videos and translated all the documents in English. The coordination of all these large groups has been possible thanks to the extensive use of a cloud platform. Due to the high number of people involved and the amount of documentation produced, the overall project has required a great and continuous organizational effort. A detailed documentation of the project workflow can be found on the dedicated page `http://appinventory.uniud.it/development/`.

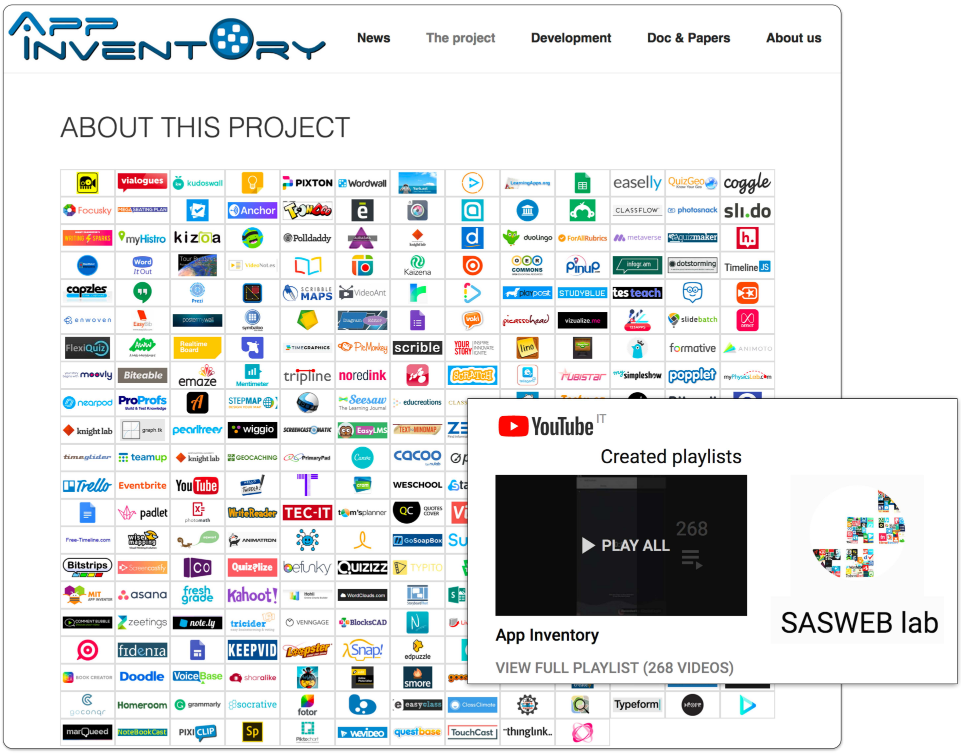

Figure 3.1: The Web page of the project, the applications' logos, and the Youtube playlist.

3.2.2 The cataloging scheme

We propose an open classification scheme which accepts user contributed use cases, since each application could have several uses also distant from those planned by its creators.

attributes	links	dimensions
Title	Application name	**Categories**
Abstract	Video-presentation	**Tags**
Public catalog availability	Video-tutorial	**Typology**
App presentation and screenshots	**Documentation**	**Complexity (1-10)**
Free plan limits	Plans & pricing	**Bloom's levels**
Artifacts' access policy	**Product examples**	**Registration policy**
Registration notes		Presence of advertising
Specific subjects		Italian language interface
Statistics and evaluation data		User comments and contributions

Table 3.1: The cataloging scheme: some attributes are links to external resources, others are dimensions which semantically connect the apps in the underlying zz-structure model. Bold indicates multi-valued attributes.

The initial items of the cataloging scheme are listed in Table 3.1 and are visible in the cards of each application (see also the specific card proposed in Figure 3.8). App data include simple and multi-value attributes, external links and dimensions which link together the apps according to different criteria, as discussed in the zz-structure-based data model (Section 3.5). The considered attributes for the applications are *title* and *abstract* respectively a short (one line) and extended (some lines) descriptions of the app, the *application name* which also represent a link to the related website, an attribution to a list of *categories* in the taxonomy presented in Subsection 3.2.3, a link to a *video-presentation* created specifically for the AppInventory project, a link to a *video-tutorial* in English and Italian language, a list of descriptive *tags* associated to the app, a flag to indicate the *availability of a public catalog* of artifact on the app website, the list of the app *typologies*, e.g. Webapp, Android app, IOS app,. . . , an extended textual *app presentation* integrated with a series of figures illustrating the app pages, a list of links to external *documentation* (Web resources documenting the app, such as support pages, presentation slides, Q&A collections, manual pages, blogs,. . .), an estimate of the *complexity* of use of the app in a 1-10 range, a description of the *limits of the free plan* offered by the app, a link to any *plans & pricing* webpage of the app (generally a comparative chart of the offered features by free and commercial plans), a list of attributions to *Bloom's taxonomy levels* supported by the app, a flag to indicate the *access* and *registration policies* of the app together with *registration notes* describing the various sign-up methods accepted by the app, a list of links to *product examples* created using the app, a list of any specific *subject* supported by the app, flags to signal the *presence of advertising* and the availability ot *Italian language interfaces*. Finally a series of *statistics and user evaluation data* and user contributions like comments and specific use-case, as illustrated in Subsection 3.4.2.

3.2.3 The purpose-based taxonomy

Having observed recurrent purposes, we mapped the applications into 3 macro-categories, as illustrated in Figure 3.2.

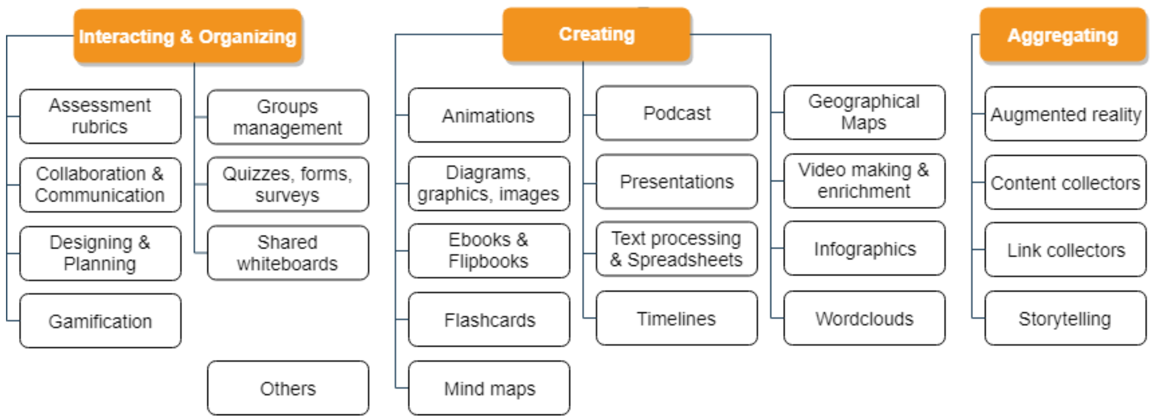

Figure 3.2: Our purpose-based taxonomy.

- *Interacting & Organizing* includes applications to manage groups, to collaborate on the same documents online, to support users in planning projects and activities, to interact in real-time on a virtual board or to collect data by surveys and quizzes.

- *Creating* includes applications that support users in building up digital artifacts, belonging to various typologies. Generally, after an initial registration, these applications offer users a personal dashboard to manage their digital products and an editing environment where to build and modify them. It is generally possible to share the artifacts by a specific url, an embed code or by directly publishing them on social platforms.

- *Aggregating* contains the applications which support users in collecting homogeneous or heterogeneous materials (for example links, images, videos, documents, maps, events) in order to semantically connect them, to keep notes about interests, to create stories, to distribute and share the resulting collections in a simple manner.

The macro-categories are structured in relative categories: 13 for the 'Creating', 7 for 'Interacting & Organizing', 4 for 'Aggregating', plus an additional generic 'Others' to capture unforeseen features. Each application often integrates various distinct features: for this reason, we have adopted a weighted attribution on 3 levels $(1/3, 2/3, 1)$ of an application to single categories in order to highlight the primary purpose compared to secondary ones.

3.2.4 Statistics on the dataset

The distribution of the 281 apps into the taxonomy is shown in Figure 3.3. The total number of the apps is greater than 281 since each application can be assigned to more than one category. Figure 3.4 shows the distributions of the apps in the dataset according to

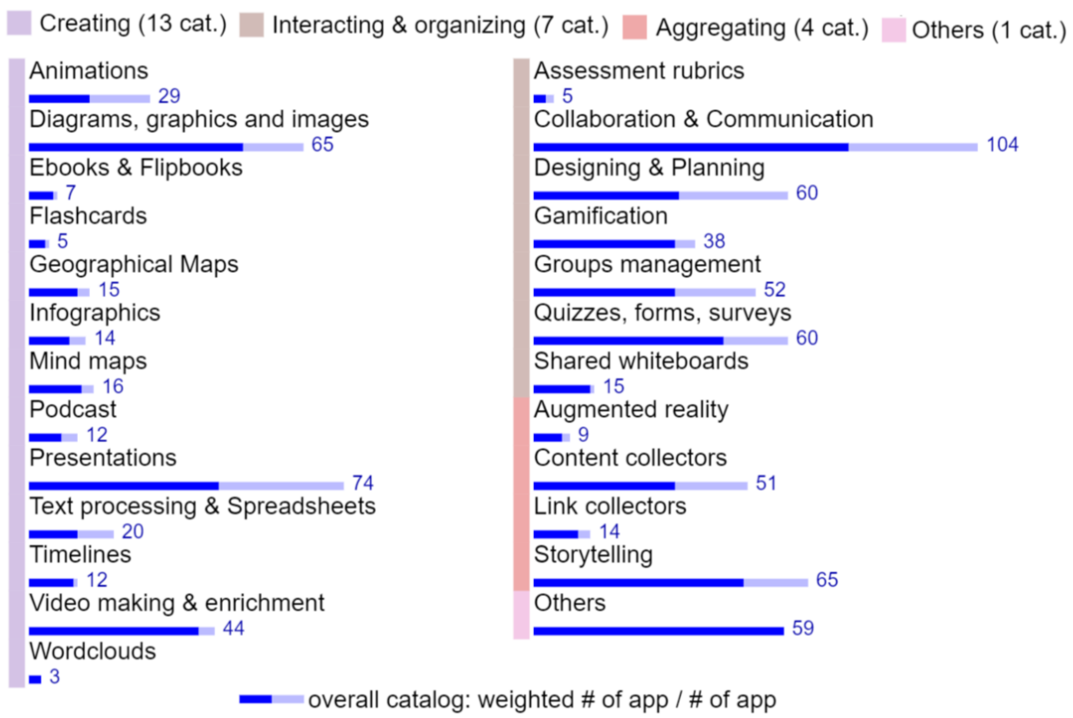

Figure 3.3: Distributions of the 281 apps into the categories and macro-categories. The lengths of the blue bars are proportional to the weighted number of applications obtained by summing the attributed weight of each application to the category. In azure are indicated the absolute numbers of applications for each category.

some of the considered features: *typologies*, *registration policy*, *public catalog availability*, *complexity of use* in a range from *1=straightforward* to *10=very complex*, *Bloom's levels* attributions, and *presence of advertising*. We observe a relatively uniform distribution of the applications over the six levels of Bloom's taxonomy. This in part reflects the versatility of the analyzed tools: for example, an application to create online presentation

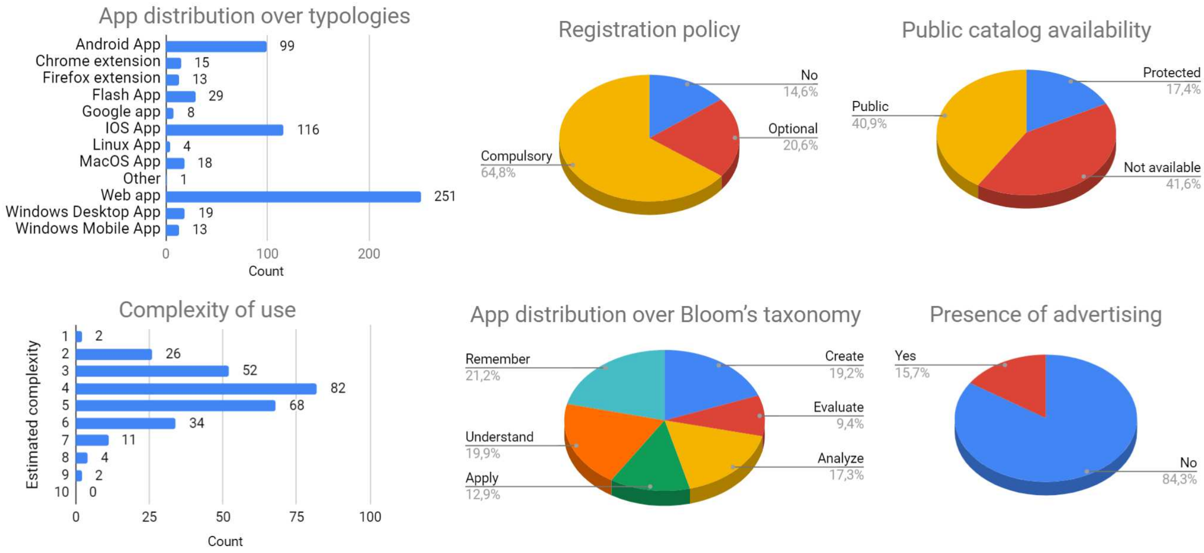

Figure 3.4: The distributions of the apps in the dataset according to some of the considered attributes.

can be used by teachers to support their students in the memorization and understanding of concepts but it also represents a tool to develop the analysis and the creativity skills when used by students to summarize a topic and create an effective presentation.

The app typology field describes the various forms in which an application is made available: due to the multiform nature of many applications, a multiple attribution to the various typologies is possible. Most of the analyzed apps are available as Webapps based on modern Web standards like HTML5, SVG, CSS3, ECMAscript, etc., and they are generally responsive and portable. A minority still adopts proprietary solutions like Flash that limit their portability to desktop devices, but there is a general tendency to progressively migrate towards Web standards: many applications provide both Flash and HTML5 versions and encourage users to choose the last one for the new creations. About 81% of the applications analyzed are app for mobile devices or are Webapps that also offer an optimized version for mobile device. Another significant aspect we took in consideration concerns the need for authentication in order to create artefacts or in general to access its functions. A large part of the analyzed applications (about the 65%) require users to register and authenticate before using them, about 20% of the applications can be used anonymously with limited features (for example without save or share functions), the remaining 15% of the analyzed apps do not provide registration and authentication procedures mainly because they offer simple services, like the generation of visual representations (e.g. wordclouds, qr-codes, etc.), the conversion of file formats or the editing of images with the direct download of the products and without the use of permanent remote resources. Regarding the *complexity of use*, an average value of 4.3 reflects a relative ease of use of the applications considered, thanks also to the continuous evolution of the user interfaces in the direction of usability.

3.3 The Web platform and its architecture

AppInventory has been implemented as a Web application based on HTML5, SVG and CSS3 W3C standard languages and the D3js [32] framework. D3 provides a powerful

DOM selection mechanism, based on declarative CSS patterns; a rich library of methods to create complex graphical representations and to act, with the same syntax, both on single DOM elements and on sets. The idea behind D3 is to strictly tie data to HTML or SVG elements realizing a so-called data-driven approach to DOM manipulation without hiding the document structure with opaque software layers. We recently experimented the D3's versatility in realizing the application VisualBib [50]. AppInventory adopts AJAX techniques to improve a user experience by avoiding full page reloads during navigation, by dynamically loading or sending on demand only small chunks of data from/to the server.

The client-server architecture of AppInventory is schematically represented in Figure 3.5 and the main components are discussed below.

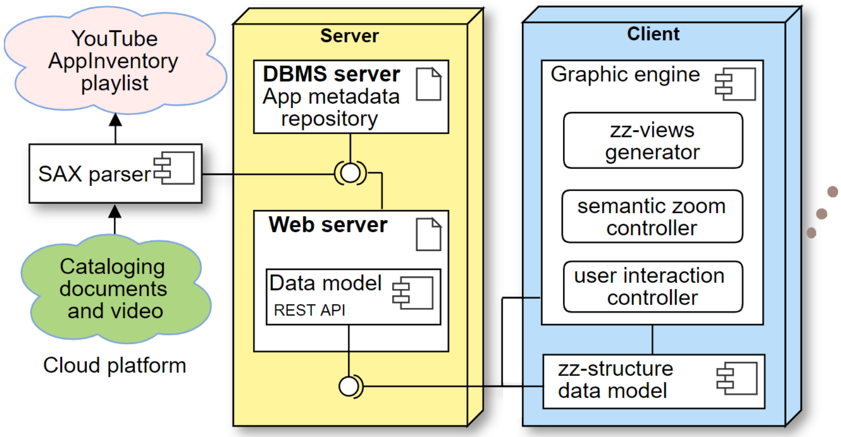

Figure 3.5: The architecture of the AppInventory framework.

The SAX parser is a Java component (Figure 3.5-left) which support the system administrators during the process of adding new data to the *YouTube AppInventory playlist* and to the DBMS (DataBase Management System), parsing the new documents that become available in the *cloud platform*, used in the cataloging phase, as described in Subsection 3.2.1. For each new documented application, the SAX parser extracts and validates all the significant metadata from the XML versions and generates appropriate SQL statements in order to add new records into the database and to establish the opportune data relationships.

The server node (Figure 3.5-center) contains the DBMS, where the data are modeled in a relational scheme implemented in a MySql server; it represents the *App metadata repository* of AppInventory. The *Web server* provides the static contents (html pages, images, scripts and stylesheets) to Web clients and manages, through the *Data model* component, the asynchronous requests for data retrieval/update, received on its REST API endpoint from the client side components of AppInventory.

The client node (Figure 3.5-right) becomes active during the application running on the browser of each user. It contains two main components, zz-structure data model and graphic engine, which manage respectively the semantic browsing mechanisms and the

graphical layout of the catalog.
The *zz-structure data model* uses a conceptual semantic model for structuring the data, the so-called zz-structures [89, 54, 60, 59, 57]. It defines and manages the zz-dimensions, described in Subsection 3.5, which semantically connect applications, categories, external items and metadata. This component manages both the static zz-dimensions, which model the pre-established relationships between items, and dynamic zz-dimensions which are created on the fly, as a result of user actions. For example, a new zz-dimension is generated during a search session to semantically connect all the found apps. Another case occurs when user composes multiple zz-ranks by an AND / OR operator, using the semantic browsing mechanism, presented in Section 3.4. This component also maintains and synchronizes with the server the dynamic data generated during the user navigation, for example, when new ratings for apps are added, new user comments are inserted or the visit and use counters associated to the apps are updated.
The *graphic engine* generates the holistic, visual, and interactive representations of the domain data and supports the semantic zoom behaviour during the navigation, revealing/hiding contextual information. It implements specific zz-views to show and connect the elements of the catalog and manages users' interaction during the exploration, offering navigation mechanism like the current rank navigation window (Figure 3.14-right) and the contextual rank selection and composition window (Figure 3.14-center). It also generates the views of the app information cards, the link to external resources associated to each app and implements some protection mechanisms, based on cookies and invisible recaptcha techniques, to avoid multiple ratings of an item by the same user and to block spam attacks.

3.4 Modelling the graphical layout

We defined the following main guidelines for the graphical layout of the AppInventory catalog, with the aim to provide innovative and usable modalities of navigation, in according to the Shneiderman's mantra [92] - *"Overview first, zoom and filter, then details-on-demand"*:

- present an initial comprehensive view of the entire repository, without exposing details of the apps;
- offer a continue zoom mechanism in order to minimize users' disorientation and let them choose the appropriate level of visualization;
- propose a semantic zoom mechanism: each item becomes visible at an appropriate zoom level in order to enhance the understanding and minimize the cognitive load;
- users can freely navigate in multiple directions, using next-previous contextual move mechanisms.

For the implementation, we evaluated alternative interactive visual representations of the data in order to offer attractive solutions to user navigation and to highlight specific data relationships. The literature on visual interfaces and languages is rich of proposals [51]; a graphical review on visual languages from 1995 to 2014 is discussed in [52], where the authors gathered and analyzed the employed visual techniques (graph-based visualization such as collaboration, co-citation, and co-word networks) and adopted geographical views, alluvial diagrams, and timelined charts. An interactive visual browser is

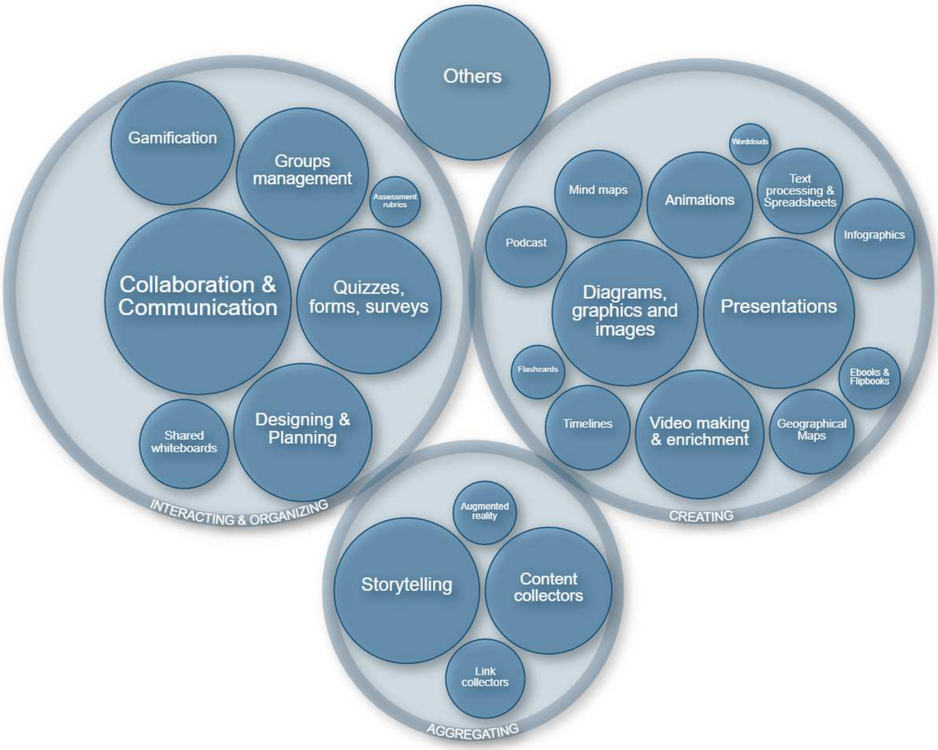

Figure 3.6: The holistic view of the catalog.

presented in [73], where the authors collected 430 different text visualization techniques. The catalog displays a card for each entry. The Web page also addresses a set of other surveys on some projects, such as BioVis [71] and SentimentVis [74] propose visual guide for data visualization techniques, in the fields of biology and sentiment analysis, respectively. Many live examples of interactive visual representations of complex data can be found in D3.js [32, 38] and Echarts [79] frameworks.

In order to represent our purpose-based taxonomy with multiple and weighted attribution of application to categories, we analyzed various solutions and finally chose to implement a multi-level version of a circle packing diagram, which we enriched with semantic zoom and browsing mechanisms. The formal description of such view in the context of zz-structures is introduced in the Subsection 3.5.1 after the definition of the zz-structure model of the catalog (see Subsection 3.5).

Figure 3.6 shows the initial holistic view of the AppInventory catalog; at the first zoom level the macro-categories are represented by separate circles, which contain the 24 categories. The size of each circle is proportional to its populousness. Zooming in, taking the focus on the category ‘Mind maps’, a new view (Figure 3.7-left) reveals the logos of the applications which populate this category. The size of each circle is proportional to the weighted attribution of an application in a single category and the gray color identifies apps that are no longer active. The next level of zooming (Figure 3.7-center) enables users to visualize, in addition to logos, the names of the apps as well as a title. In Figure 3.7-right, we zoomed on the Mindmeister app by clicking on its name: further navigation elements appear to enable navigation towards similar applications in the same category (left and right arrows) and the other categories to which the app belongs (the four buttons visible on the bottom). In addition, two buttons appear on the top: the

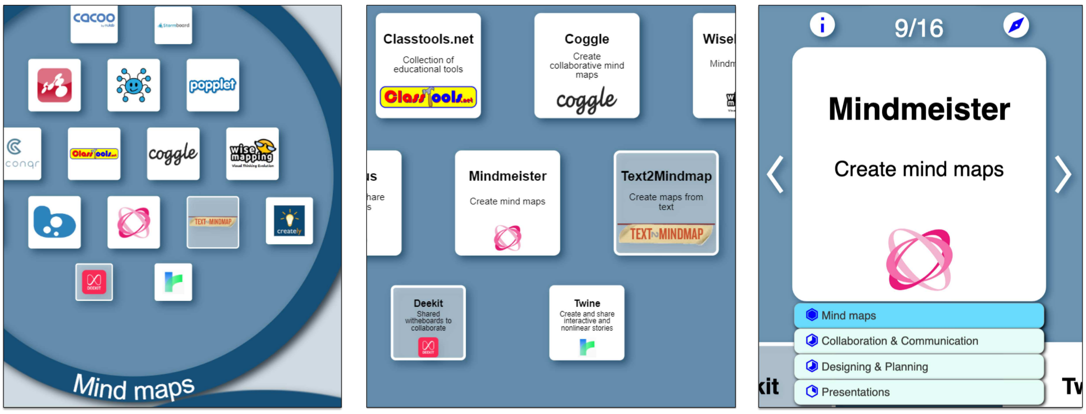

Figure 3.7: Zooming in the view, the apps' logos appear (left); additional zooming in makes visible names and titles (center); clicking on an app, appear new details (right).

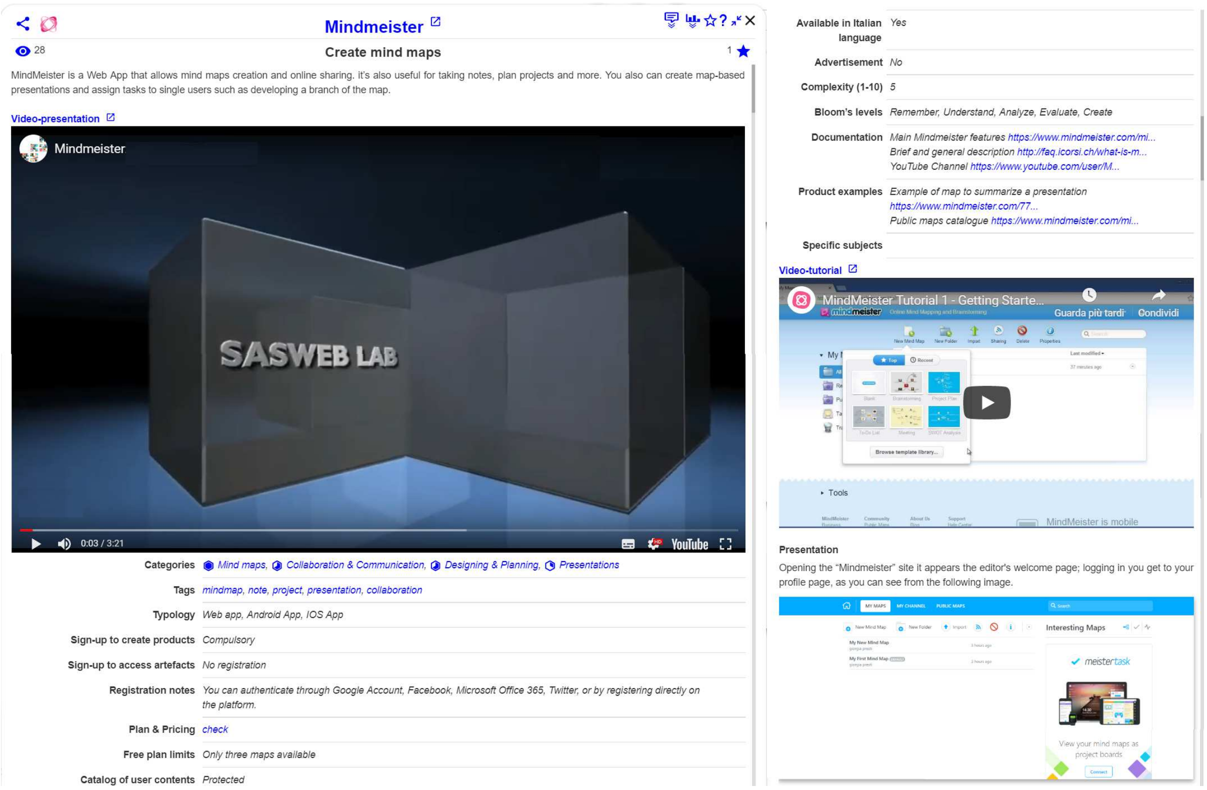

Figure 3.8: A partial view of the application's card.

"compass" button opens the contextual rank selection window, discussed in Section 3.5, while the "i" button opens the detailed information card of the app, partially visible in Figure 3.8, where, on the top-right, there are six icons to:

- open the comment section of the app, discussed in next Subsection 3.4.2;
- open the rating section of the app, discussed in next Subsection 3.4.2;
- visualize if the app is known and used by the user;

- activate the guided tour related to this window;
- enlarge the window;
- close the window.

Next to the subtitle, the number of the visits and the number of users who declared to know and use the app, are visible. These counters are increased at most once per user's session. The information card visualizes all the fields proposed in the cataloging scheme (see Figure 3.2.2), and among the others:

- a short description of the main purpose of the app;
- an original video presentation of the app, recorded by students of the work group; most of videos are accompanied by English subtitles: the translation work is still ongoing;
- a list of fields to describe the app according to the different taxonomies;
- a third-party video-tutorial in the currently selected language (Italian or English);
- a review of the app that describes its main features through text and images in order to give the user the opportunity to evaluate the adequacy of the app with respect to his/her goals.

3.4.1 Basic and advanced searches

AppInventory provide a basic and advanced search (see Figure 3.9). The simple search

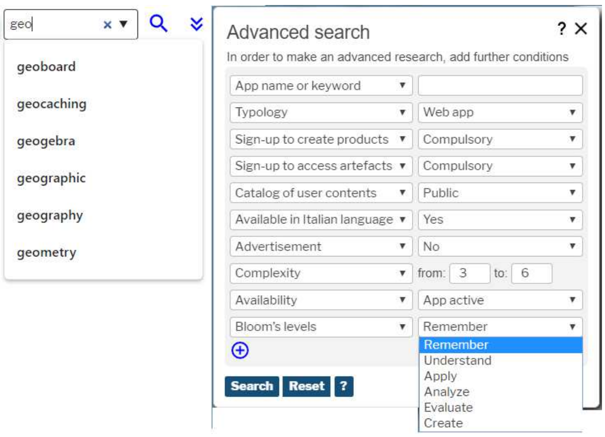

Figure 3.9: The basic search bar (left) and advanced search form (right).

enables users to find applications specifying part of the name or some keywords. While typing, users receive real-time suggestions; this is achieved through Ajax calls to the server which generate opportune hints by matching app names and tags using Levenshtein string distances in the matching algorithm. Figure 3.9 (right) shows all the search criteria accepted by the advanced search form. As discussed in Section 3.5 the result of a research can be explored and composed with other zz-dimensions of the catalog in order to refine the result and restrict the navigation set. After a search, the main view of the catalog highlights the position of all the found apps in the categories, offering an overall representation of the distribution of the results.

3.4.2 The rating scheme

In this new version of the catalog, we introduced the opportunity for the users to interact with the platform, enabling them to rate the applications, leave personal comments, annotate them as known and used, suggest new use cases or new applications to add to AppInventory. Figure 3.10 visualizes the possible ratings; each user can rate four features

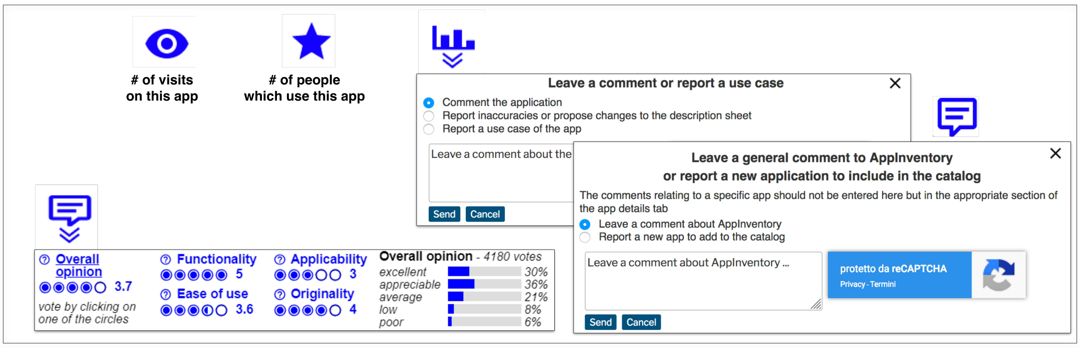

Figure 3.10: Rates, comments, suggestions.

of any app, and express a general opinion in a 5-Likert scale:

- *functionality*: versatility of the app or the richness of the features provided;
- *applicability*: adaptability of the app to multiple contexts and tasks;
- *ease to use*: usability and the intuitiveness of the user interface;
- *originality* is referred to the features provided and/or the technical adopted solutions;
- *overall opinion* is the overall degree of appreciation of the app.

For each features, the user can see the rating and its distributions on the 5-Likert scale. In addition to ratings, users can leave five different types of comments in two contexts:

- local to a single app: comments to the app, suggestions of original use cases or reports of inaccuracies / proposals of changes in the description card;
- global: comments about AppInventory or suggestions of new apps to add in the catalog.

3.5 The zz-structure-based data model and the semantic browsing

The data model uses a conceptual semantic model for structuring the data, the so-called zz-structure [89, 54, 60, 59] introduced in Chapter 1.

A multimedia catalog of applications can be thought as a multigraph where vertices are app items, categories, comments, evaluation sheets, ... connected to each other along multiple dimensions to model the semantic relations between them. In order to formally introduce the zz-structure model of AppInventory we propose a definition of a catalog of applications.

Definition 3.5.1. The AppInventory multimedia catalog of applications - The multimedia catalog of applications *AppInventory* is as a zz-structure, where

- the set of vertices $V = \{A,\ C,\ MC,\ DE,\ EV,\ COM,\ TYP,\ BL,\ TAG\}$ is composed by:
  - the set A of n applications ;
  - a set C of 25 categories and a set MC of 3 macro-categories ;
  - n detail cells DE and evaluation data cells EV associated to each application;
  - a set COM of user comments and contributions;
  - a set TYP of 12 app typologies ;
  - a set BL of 6 labels related to the Bloom's taxonomy levels;
  - a set TAG of m tags words.

  The composite vertices in DE contain, for each app, the data listed in the first two columns of the cataloging scheme of Table 3.1 except statistic and evaluation data. These last are collected in the EV composite vertices where, for each of five features in the rating scheme, the number of votes for each level in the 5-Likert scale are stored, together with the views and the uses counters related to the app.

- $D = \{d.c_1, \ldots, d.c_{25}, d.mc_1, \ldots, d.mc_3, d.mc, d.typ_1, \ldots, d.typ_{12}, d.bl_1, \ldots, d.bl_6,$ $d.tag_1, \ldots, d.tag_m, d.complexity, d.adv, d.ita, d.comments, d.details, d.evaluations,$ $d.app\text{-}l, d.pp\text{-}l, d.videop\text{-}l, d.videot\text{-}l, d.samples\text{-}l, d.docs\text{-}l\}$
  where:
  - $d.c_1, \ldots, d.c_{25}$ identify the dimensions which connect the applications to categories. The name of the category is the maincell of the corresponding dimension; the attribution of each application to a category dimension is multiple and weighted on 3 levels: $1/3, 2/3, 1$;
  - $d.mc_1, \ldots, d.mc_3$ identify the dimensions which connect the categories belonging to the three macro-categories $mc_1, \ldots mc_3$ whose complete name is the maincell of the corresponding dimension;
  - $d.mc$ identifies the dimension which connects the three macro-categories in MC and the category *others*;
  - $d.typ_1, \ldots, d.typ_{12}$ identify the dimensions which connect the applications of typology $typ_1, \ldots typ_{12}$; the complete name of the typology is the maincell of the corresponding dimension. Examples of typologies are Webapp, Android app, IOS app, $\ldots$;
  - $d.bl_1, \ldots, d.bl_6$ identify the dimensions which connect the applications associated to the 6 bloom levels $bl_1, \ldots bl_6$. The name of each level is the maincell of the corresponding dimension;
  - $d.tag_1, \ldots, d.tag_m$ identify the dimensions which connect the applications respectively tagged with $tag_1, \ldots tag_m$; each tag is the maincell of the corresponding dimension;
  - $d.complexity$ identifies the dimension which connects the applications having in common the same level of complexity of use. This dimension groups apps in 10 ranks $(1, \ldots, 10)$, marked with the respective numeric value;

- *d.adv* identifies the dimension which connects the applications having in common the presence/absence of advertising in their interfaces; it groups applications in two ranks marked with the 'yes/no' labels;
- *d.ita* dimension connects the applications making available user interfaces translated in Italian language; it groups applications in two ranks marked with the 'yes/no' labels;
- *d.comments* dimension connects each application with the list of related user comments and use-case contributions;
- *d.details* dimension, constituted by parallel ranks, connects each application to a composite cell in the DE set containing information details about the applications and links to external information;
- *d.evaluations* dimension, constituted by parallel ranks, connects each application to a composite cell in the EV set containing all the user evaluation data and statistics;
- *d.app-l, d.pp−l, d.videop-l, d.videot-l, d.samples-l, d.docs-l* identify the dimensions, constituted by parallel ranks, connecting external resources, specifically:
  * *d.app-l* links the website of the app;
  * *d.pp-l* links the page of the app website related to the plans & pricing description page of the app, if any;
  * *d.videop-l* links the presentation video(s) of each app hosted in the Youtube channel of Sasweb laboratory;
  * *d.videot-l* links the tutorial video, if any, hosted on Youtube or Vimeo cloud platforms;
  * *d.samples-l* links to multiple external Web resources representing examples of products / artifacts created with the app;
  * *d.docs-l* links to multiple external Web resources documenting the app;
- $d.found$ is a dimension which connects all the application found after a simple or advanced search. It represents a dynamic dimension since its rank is redetermined as a result of user actions.

Figure 3.11 shows a representation of a small part of the zz-structure model in the form of edge-coloured multigraph including:

- three applications a_1 (classmill), a_2 (issuu), and a_3 (dropr);
- six categories $c_1, \ldots, c_6$, respectively Collaboration and Communication, Presentations, Ebooks & Flipbooks, Content Collectors, and Storytelling;
- three macro-categories mc_1, mc_2, mc_3,respectively Interacting & organizing, Creating, and Aggregating;
- one typology typ_1 (Webapp);
- four Bloom levels bl_1, bl_2, bl_4, bl_6, respectively Remember, Understand, Analyze, and Create;
- three tags tag_1 (Lesson), tag_2 (Book), tag_3 (Collect);

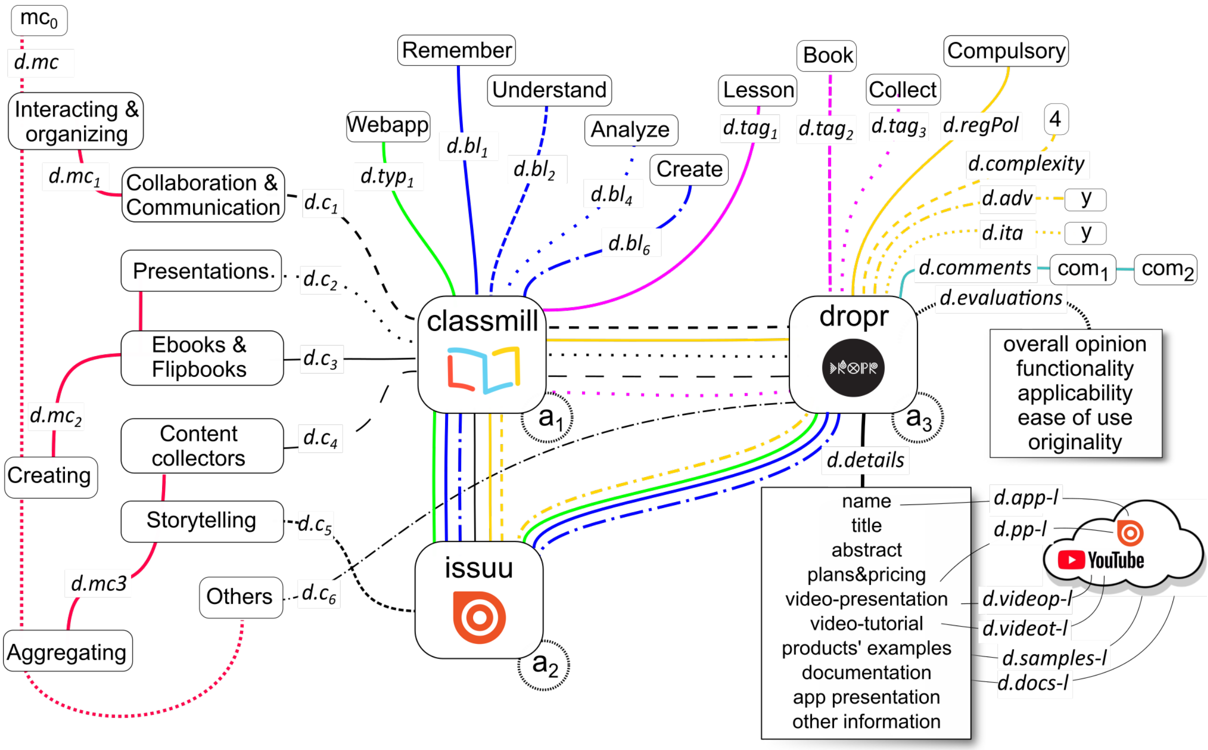

Figure 3.11: A representation of a small part of the zz-structure model adopted in AppInventory.

- one detail cell de_3 for the application a_3;
- one evaluation data cell ev_3 for the same application a_3.

The lines of different colors and styles highlight part of the defined ranks in the involved dimensions.

By combining these dimensions through AND and OR operators is possible to dynamically generate new dimensions for the contextual exploration of the neighborhood of a certain app.

3.5.1 Zz-views in AppInventory

The model of the main view of AppInventory is illustrated in Figure 3.12 and consists of a 3-level nested bubble-view applied respectively to the dimensions $d.mc; d.mc_1, \ldots, d.mc_3, c_6$, and $\{c_i\}, i \neq 6$. Consistently with the adopted taxonomy, the cell c_6, corresponding to the category *others*, is placed at the same level of the macro-categories $d.mc_1, \ldots, d.mc_3$. In order to simplify the view, the maincell mc_0 of the dimension $d.mc$, representing the root node of the taxonomy, is omitted in the actual view, visible in Figure 3.6. According to the bubble-views model, described in Subsection 1.3.6, application cells belonging to more than one category are duplicated in the view in order to avoid overlapping of the category bubbles. As illustrated in Section 3.4 the view is zoomable and draggable and a semantic zoom mechanism provides to progressively expose details, such as application icons and titles, at the opportune zoom levels. When an application is focused by clicking on it, some additional elements become visible in order

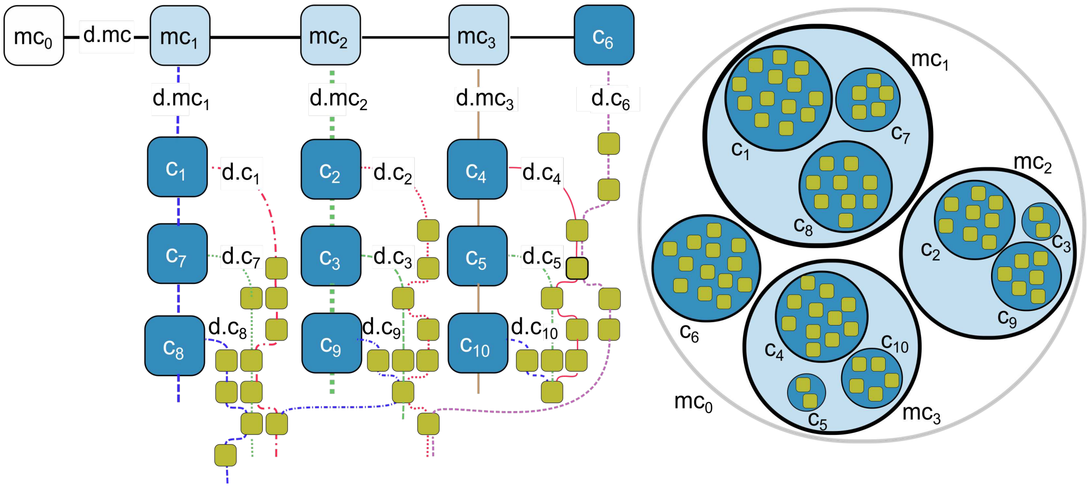

Figure 3.12: The 3-level nested bubble-view model adopted for the main zz-view of the catalog. The portion of the zz-structure (left) is rendered through a chart (right) where the macro-categories are displayed by light-blue circles, the categories by blue circles and applications by green squares.

to semantically connect the item to the others in the system. For example, Figure 3.7 (right) shows a left and right cursor to move along current rank (e.g. current category dimension), an "i" icon to access the app information card (Figure 3.8) through the *d.details* dimension, a small "compass" icon for contextual exploration and a list of the headcells along the involved c_i dimensions (for Mindmeister the five categories of this application: *Mind maps*, *Collaboration & Communication*, *Designing & Planning*, and *Presentations*). Further information about the app can be accessed in the view of Figure 3.8 where all details are presented and the exploration of the connected ranks can be activated by clicking on Web links marked with the headcell of corresponding dimension (e.g. a category, a tag, ...) or with an opportune label.

Another important zz-view of AppInventory is generated when clicking on the "compass" button, located on the top-left of the main page of the catalog. It consists of a simple list-view of the current navigation set of applications, typically one of the c_i dimensions corresponding to the last visited category in the catalog. This rank is re-determined when user clicks on another category, performs a search or change criteria during contextual navigation, as we discuss below. Figure 3.13 shows three instances of list-views with different sorting criteria and dimensions selected: the first two refer to the same dimension and different item orders, the third show the navigation in the set of found apps, after a search operation. Changing the sorting criteria, specific information appear at the left of each app name like the number of visits or uses, the app complexity or the mean score for the selected evaluation features. The color of the list depends on the currently selected dimension: *white* is associated with the categories dimensions, *red* with the *d.found* dimension, *green* with *d.complexity* dimension, different shades of *purple* with the $d.bl1, \dots, d.bl6$ dimensions, *light blue* with the *d.ita* dimension, *orange* with the *d.adv* dimension. The *yellow* color is used to indicate a composition of multiple dimensions by *and/or* operators, as discussed below. The listed colors are also used to outline the application items in the bubble-view in order to remind users about the current selected dimension.

Figure 3.13: Three instances of list-views: the first two (left and center) are related to the same dimension (category "Mind maps") but with different sorting criteria applied and the third (right) showing the result of a search of "notes" which redefines the dynamic dimension $d.found$ colored in red. On the top are visible: the cardinality of the current navigation rank, the active dimension or combination of them, the list of available sorting criteria and the list of apps with additional information related to the chosen sorting criterion.

The navigation dimension can be changed by clicking on the "compass" button, located on the top-right of each application (as shown in Figure 3.7-right): a partial list of the dimensions involving the selected application will appear, as shown in Figure 3.14-left. In the specific example, the user initially searched for the keyword "notes" and

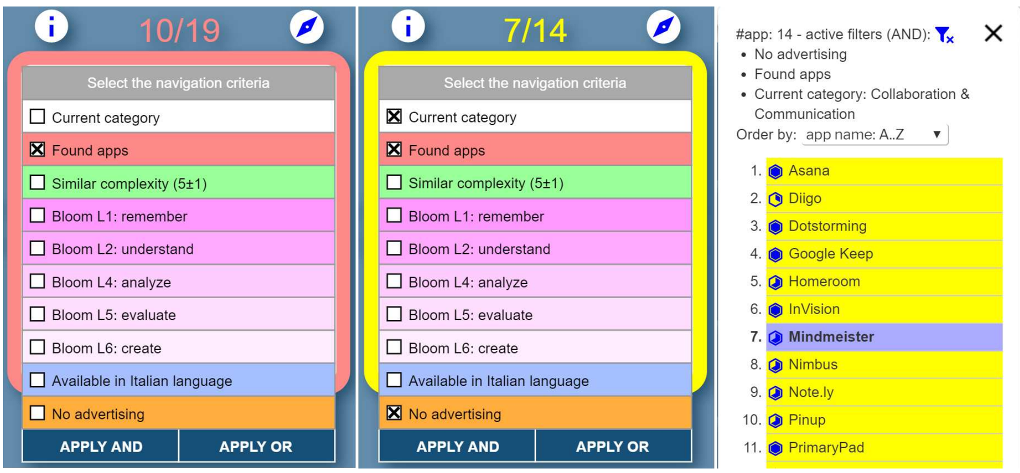

Figure 3.14: The rank selection window of the app Mindmeister after searching the keyword "notes" (left). Defining of a new dynamic rank by composing in AND three ranks (center). The navigation set of the dynamic rank (right).

the result is a set of 19 applications connected along the $d.found$ dynamic dimension.

Users can change the navigation dimension in order to find similar apps according to a certain criterion or compose multiple dimensions: in the example of Figure 3.14-center, “Current category”, “Found apps”, and “No advertising” dimensions are composed in AND obtaining a rank of 14 apps. The user can browse the obtained dynamic rank in the usual way, using the arrows or the generated list-view, shown in Figure 3.14-right.

All the views are managed by the graphic engine module which interacts with the App metadata repository through an intermediate data representation level, called data model, as discussed in previous Section 3.3.

3.6 Data analysis tools

In order to analyze and compare data about applications and categories, AppInventory provides some data analysis tools in the form of interactive graphical charts. The charts panel is generated by clicking the “Radar” icon visible in Figure 3.15 (left) and includes two radar charts (*app data* and *categories*) and a bar chart *app distributions*, described in the following subsections. Radar charts were chosen for the compactness in the visualization of multiple series of multidimensional data and a particular care has been devoted to integration in the catalog to ensure the propagation of user interactions between charts and the bubble/list views of AppInventory. All the charts are generated by specific functions implemented over the D3 framework [32] and SVG standard.

3.6.1 Radar chart of the applications

This radar, visible in Figure 3.15 (left), summarizes 16 features of the apps, grouped in four sectors:

1. *App attributes*
   - *sign-up* describes the registration policy of an app with 3 possible values: *none*, *optional*, *compulsory*;
   - *complexity* of the app, attributed in a range 1-10;
2. *Bloom’s levels* describe the attributions of an app to the six Bloom’s taxonomy levels;
3. The mean values, in a range 1-5, of the *scores attributed by users* on five evaluation features:
   - *originality*
   - *ease of use*
   - *applicability*
   - *functionality*
   - *overall opinion*
4. Users’ *counters*:
   - *# of visits*
   - *# of votes*

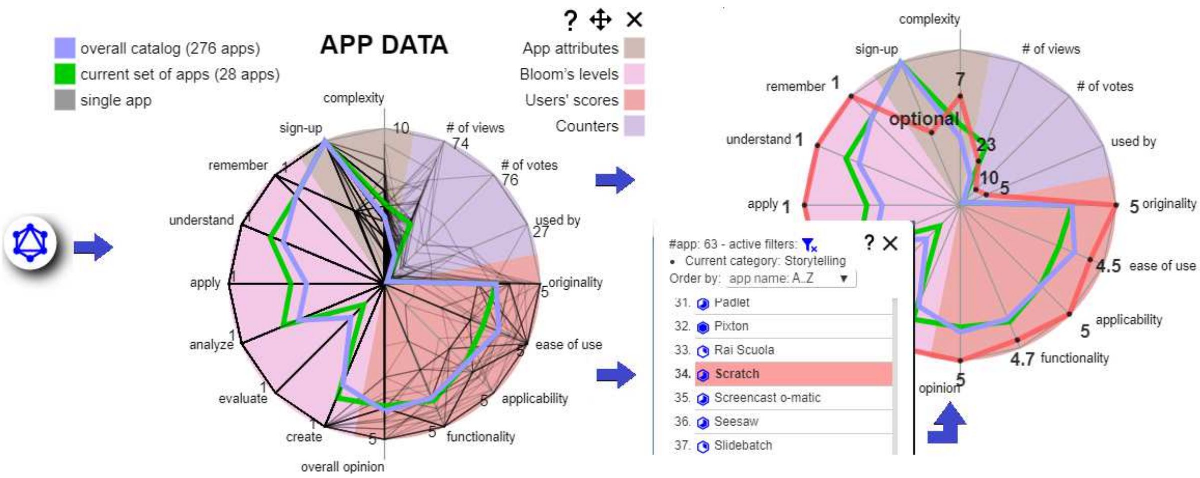

Figure 3.15: The interactive radar chart of the applications: metrics are grouped in four sectors: "app attributes","Bloom's levels","scores attributed by users", "counters". User can interact with the radar and highlight the data of a specific app (right).

- *# of uses*

The radar chart displays a curve for each application of the catalog (in gray) and two summary curves: the mean values calculated on the overall catalog (in blue) and the mean values calculated on the apps in the current navigation set (in green). Users can interact with the charts in many ways:

- by moving the mouse over a specific curve: the other application curves are temporarily hidden and the values, for each of the 16 features, are visualized;
- by moving the mouse over the items of the two legends: the corresponding curve or sector are highlighted;
- by moving the mouse over the labels of the axes: the scale relative to the feature and the corresponding grid become visible. The range of the scale can be fixed or dynamic, e.g. the maximum number of visits, votes,. . . ;
- by clicking on a specific app curve: the corresponding application in the catalog is selected.

Furthermore, if the list-view of the catalog is visible, any interaction (mouse over and click) on the list of items highlights the corresponding curve in the chart. The radar chart of the applications helps user to find answers to the following questions:

- What are the values of the metrics for a certain app?
- For each metric, what position does an app occupy compared with all the apps in the catalog?
- For each metric, what position does an app occupy compared with a specific subset of apps in the catalog?

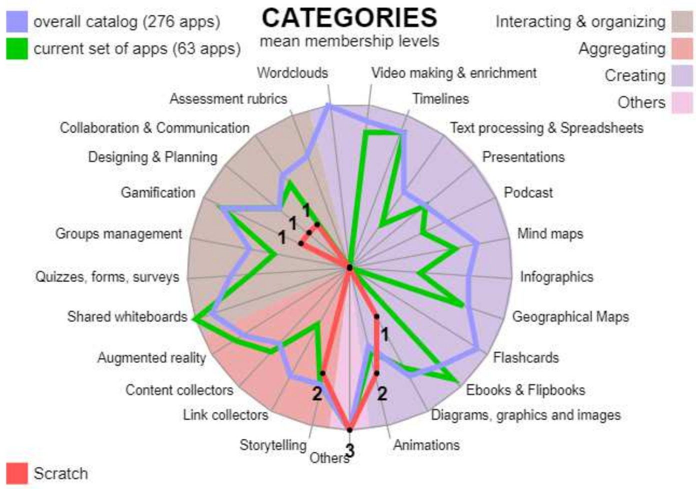

Figure 3.16: The interactive radar chart of the categories: the 3 curves shows, for each category, respectively the mean membership levels of the apps of the catalog (blue path), the mean membership levels of the apps of the current navigation set (green path) and the membership levels of the selected app (Scratch in the example) to the categories.

3.6.2 Radar chart of the categories

Figure 3.16 shows the radar chart relative to the categories. It contains three curves describing respectively:

1. the mean membership levels of the apps of the catalog to each category (blue curve);
2. the mean membership levels of the apps of the current navigation set to each category (green curve);
3. the levels of attribution, in a range 1-3, of the selected app to each category (red curve).

This chart does not give information on the number of applications belonging to each category but on the characterization of the various categories: high values for a category (e.g. value 3 for *wordclouds* relative to all the apps of the catalog) indicate a strong characterization of its applications (all apps in the *wordclouds* category have the maximum membership level). Lower values (e.g. 1.5 for *Animations*) indicate an average weak attribution of the apps of that category.

3.6.3 Distribution of apps in the categories

The chart in Figure 3.17 is generated by clicking the link labeled "app distribution" placed below the radar chart of Figure 3.16. The bar chart shows:

- the names of the macro-categories with the number of categories (top-left, top-right);
- the list of the 25 categories of the catalog grouped by macro-category and for each of them:

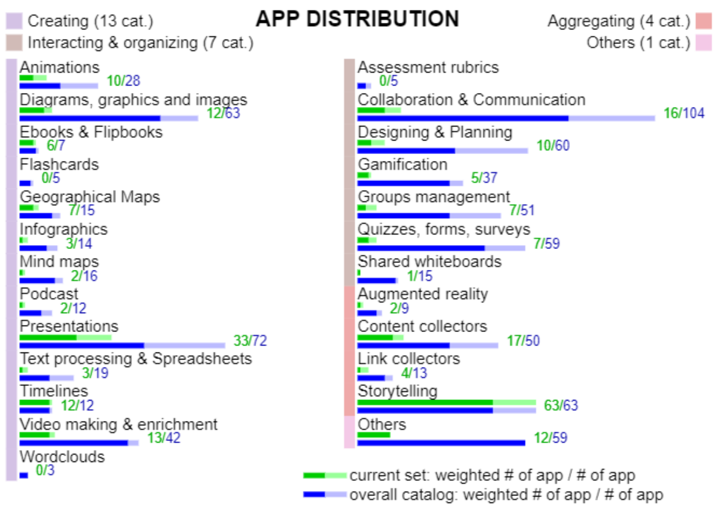

Figure 3.17: The bar chart showing respectively the distribution of the apps of the entire catalog into the various categories (blue bars) and the apps of current navigation set (green bars, related to *Storytelling* category in this example). The lighter bars indicate the absolute numbers of apps while the darker indicate the normalized ones, taking in account the membership levels.

- a light-blue bar of length proportional to the number of applications in that category; the numeric value is reported in the denominator of the fractions on the right;
- a blue bar of length proportional to the weighted number of applications in that category; it is calculated adding the membership levels of all the apps, normalized in a range 0-1; the numeric value is reported in the numerator of the fraction appearing when moving the mouse over the bar;
- a light green bar of length proportional to the number of applications of the current navigation set in that category; the numeric value is reported in the numerator of the fraction on the right;
- a green bar of length proportional to the weighted number of applications of the current navigation set in that category; it is calculated adding the membership levels of all apps, normalized in a range 0-1; the numeric value is reported in the numerator of the fraction appearing when moving the mouse over the bar.

3.7 The guided tour of AppInventory

In order to support users in discovering the features of the AppInventory platform, we realized a guided tour which presents, in the selected language (English or Italian), the single elements of the application. The tour can be started by clicking on the “?” icon and is currently organized in 64 steps: a small contextual window appears near the interface element to be described. Users can go forward or backward in the tour by clicking the next/prev. buttons: this causes zooming and panning of the view and the possible opening of the application panels to be described (e.g. the advanced search panel, the radar panel, . . .). In order to access the part of the tour dedicated to specific features, the a “?” icon has been included in some of the panels provided by the platform

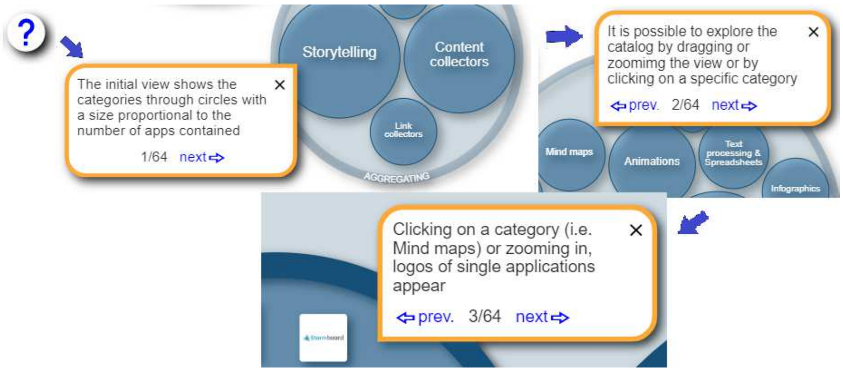

Figure 3.18: The guided tour is started by clicking the "?" icon visible on the left. A contextual window appears containing a description of the current element and prev./next buttons to move backward/forward in the tour, eventually zooming and panning the view and possibly opening of appropriate application windows (e.g. the advanced search panel, ...).

(the list-view, the information cards of the apps, the radar charts panel, the advanced search panel) enabling user to start a contextual guided tour of a specific feature of AppInventory. Formally the guided tour can be modeled as a zz-structure built on the top of zz-views elements of the user interface. Figure 3.19 illustrates such zz-structure

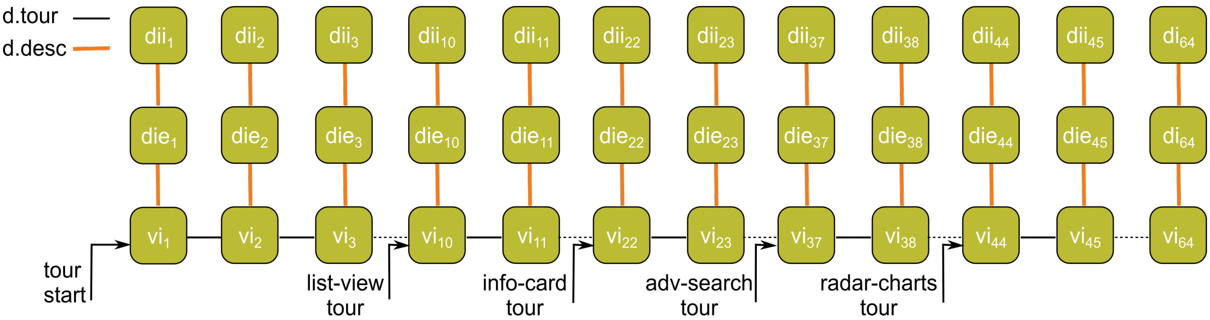

Figure 3.19: The zz-structure to model the guided tour of AppInventory. The items of views vi_k are the elements that form the user interface. Each of them is documented by the description items di_k connected to vi_k along the $d.desc$ dimension. The tour is the rank along the $d.tour$ dimension linking all the cells vi_k. It can be traversed from the beginning or from some fixed position corresponding to specific application panels.

consisting of a set of vertices V=$\{VI, DI\}$

- a set VI of elements of the user interface (views-items) vi_k;
- a set DI of description items containing the explanatory text in English (die_k cells) and Italian language (dii_k cells) of the corresponding vi_k items;

and dimensions $D = \{d.desc, d.tour\}$

- $d.desc$ links each view-item to the corresponding description cells thus defining a set of $|VI|$ parallel ranks;

- *d.tour* links in the appropriate sequence all the $|VI|$ view-items.

The guided tour is therefore the rank along the *d.tour* dimension which links, in the right sequence, all the cells vi_k. The tour can be visited starting from the beginning or from some fixed position corresponding to specific application features: the list-view, the information cards of the apps, the radar charts panel, the advanced search panel.

3.8 System evaluation

In order to evaluate the impact of our visual catalog and state that the new graphical layout and semantic browsing mechanism are usable and appreciated by users, we carried out two studies: a preliminary qualitative evaluation of the AppInventory platform, discussed in next Subsections 3.8.1-3.8.3, and a comparative evaluation with two similar tools, discussed in Subsection 3.8.4. Both the studies was carried out before the development of the data analysis tool described in Section 3.6.

3.8.1 The preliminary qualitative study

Participants The first study involved a sample of 53 persons (31 F, 22 M) who participated on a voluntary basis to a seminar for the presentation of the new platform and to the next workshop session. The age of the participants was distributed between 20 and 70 years, with a mean of 47.8 and a standard deviation of 13.7, the declared profession was teacher/researcher (70%), student (17%) or other (13%). Among teachers, 20.5% were from primary school, 72% from high school, 5% from universities and 2.5% from other schools.

Procedures and apparatus During the workshop and before the study, in order to become familiar with AppInventory and its features, the participants were asked to perform:

- a list of 9 activities organized in:
    * 4 tasks - the navigation through categories, using the cursors to move between apps and the “compass” button to change navigation criteria;
    * 1 task - the access to the information of some app;
    * 1 task - the use of the simple and the advanced search;
    * 1 task - the marking of known/used applications;
    * 1 task - the access to the rating section of the well known apps for inserting personal scores on the five evaluation criteria;
    * 1 task - the access to the comment section of the app in order to enter any comments on the application, to report inaccuracies in the information or to share a use case of the app;
- a set of 5 search operations and validate the results.

All participants were provided with a PC of the same type and were free to choose the browser to use for the test: the chosen browsers were Google Chrome (89%), Mozilla Firefox (9%), Microsoft Edge (2%).

Project The study was organized as single factor qualitative study; in addition to some initial data, participants were asked to fill:

- a SUS (System Usability Scale) questionnaire [40] in order to evaluate the perceived usability level of the application; the results are described in next Subsection 3.8.2;
- a questionnaire of 21 questions on four aspects of the platform: user layout, semantic structure, navigation and research mechanisms and user contribution features; the results are described in next Subsection 3.8.3.

3.8.2 Usability evaluation

In order to evaluate the general perceived usability of the application we ask the sample to fill a standard SUS questionnaire. The SUS value was computed, for each participant, with the formula
$SUS = \left(\sum_{k=0}^{4} (A_{2k+1} - 1) + \sum_{k=1}^{5} (5 - A_{2k})\right) * \frac{100}{40}$.
where A_i is the value (from 1 to 5) of the answer to the Q_i question.

Metrics	Value
Sample size	53
Min value	50.0
1st Qu.	72.0
Median	82.0
Mean	79.2
Std. dev.	13.0
3rd Qu.	90.0
Max value	100.00

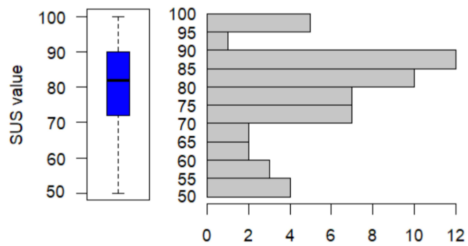

Figure 3.20: The SUS distribution (left), boxplot representation (center), and frequencies on the range 50...100 (right).

The distribution of SUS is summarized in Figure 3.20, while Figure 3.21, reports the distribution of the answers to each SUS question:

Q1 I think that I would like to use this system frequently;

Q2 I found the system unnecessarily complex;

Q3 I thought the system was easy to use;

Q4 I think that I would need the support of a technical person to be able to use this system;

Q5 I found the various functions in this system were well integrated;

Q6 I thought there was too much inconsistency in this system;

Q7 I would imagine that most people would learn to use this system very quickly;

Q8 I found the system very awkward to use;

Q9 I felt very confident using the system;

Q10 I needed to learn a lot of things before I could get going with this system.

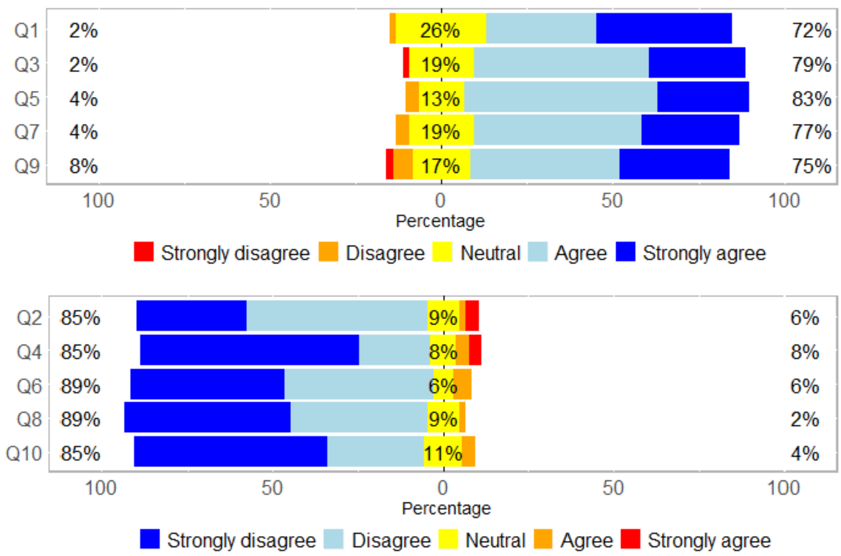

Figure 3.21: The distributions of the answers to the odd, positive tone, SUS questions (top) and to the even, negative tone, ones (bottom). In the second plot, the color scale has been reversed to map, as in the first plot, positive values to azure/sky colors.

In order to minimize acquiescence response biases the questions have an alternate tone: positive for the odd ones and negative tone for the even ones. Despite the particularity and novelty of the visual and navigation adopted solutions, the perceived usability, being SUS mean = 79.2 and median = 82 (see Figure 3.20), is between 73 = good, and 85 = excellent [40]. This first result is satisfying.

3.8.3 Analysis of specific aspects

Besides the general usability we gathered the users' opinions about four specific aspects of the platform, through a set of 21 questions, declined in positive and negative tones:

- user layout (UL1-UL5 questions);
- semantic structure (SS1-SS5 questions);
- navigation and research mechanism (NR1-NR4 questions);
- user contributions (UC1-UC7 questions).

User layout. The first group of 5 questions was about the user interface:

UL1 I appreciate the presence of an overview of the catalog;

UL2 I find distracting to zoom and drag for exploring the catalog;

UL3 I think that the AppInventory graphic layout offers innovative elements;

UL4 I consider the adopted graphical layout less effective than a traditional one;

UL5 Overall, I appreciated the graphical layout of AppInventory.

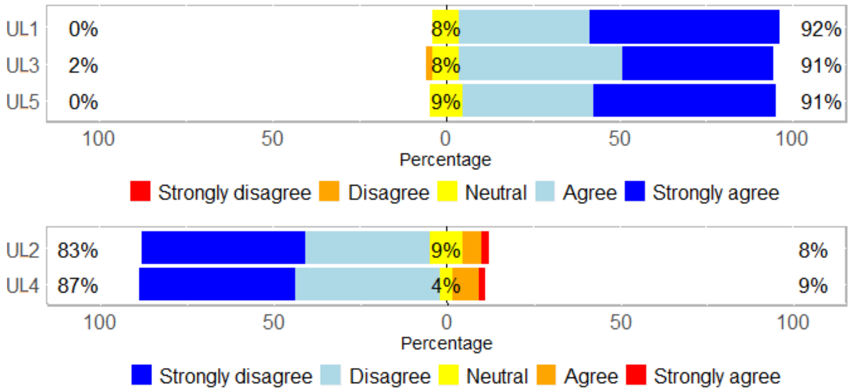

Figure 3.22: The distributions of the answers to positive tone (top) and negative tone (bottom) questions relative to the User Layout (UL) aspects. In the second plot, the color scale has been reversed to map, as in the first plot, positive values to azure/sky colors.

Figure 3.22 provides the distribution of the answers to each positive tone question (top) and negative tone question (bottom). It emerges an almost complete appreciation of the adopted interface - question L1 (92% positive, 8% neutral, 0% negative responses) and of the presence of the catalog overview - question L5 (91% positive, 9% neutral, 0% negative responses). With respect to the use of the zoom and drag for the navigation and the effectiveness of the adopted graphical layout compared to more traditional ones (questions UL2, UL4), we gathered slightly lower, but largely positive, values (on average 85% positive, 6.5% neutral, 8.5% negative).

Semantic structure. With reference to the semantic structure of AppInventory, we asked the sample to answer the following questions:

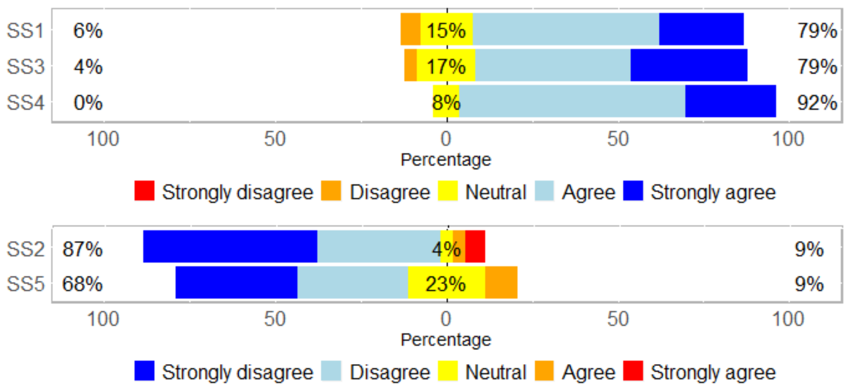

Figure 3.23: The distributions of the answers to positive tone (top) and negative tone (bottom) questions relative to the Semantic Structure (SS) aspects. In the second plot, the color scale has been reversed to map, as in the first plot, positive values to azure/sky colors.

SS1 I think appropriate the categories used in AppInventory;

SS2 I do not consider useful to assign an application to multiple categories;

SS3 I believe that the weighted attribution of an application in a category is an appropriate choice;

SS4 I found the information card of the app sufficiently complete and detailed;

SS5 The presence of a video presentation for each app is, for me, of secondary importance.

Figure 3.23 shows the distribution of the answers to the subset of positive tone question (top) and negative tone questions (bottom). Some aspects emerge: an almost complete (92% positive, 8% neutral, 0% of negative responses) appreciation of the completeness of the data provided in the apps' information cards (question SS4); a general agreement about the choice of categories and the introduction of multiple and weighted attribution of the app to them (questions SS1-SS3, on average: 82% positive, 12% neutral and 6% negative responses); a primary importance attributed to video presentations (SS5: 68% positive, 23% neutral, 9% negative responses).

Navigation and search mechanisms. The next four questions investigated about the effectiveness of the navigation and search mechanisms:

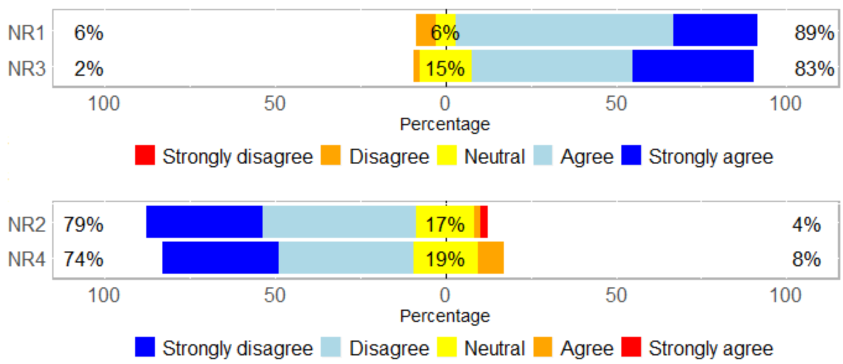

Figure 3.24: The distributions of the answers to positive tone (top) and negative tone (bottom) questions relative to the Navigation and Research (NR) features. In the second plot, the color scale has been reversed to map, as in the first plot, positive values to azure/sky colors.

NR1 I found understandable and functional the basic and advanced search mechanism;

NR2 I do not consider effective the forward / backward navigation mechanism between apps;

NR3 I consider important to visualize the contextual list of apps through the general "compass" button, located at the top-left of the graphical layout;

NR4 I think it is of little use to select a new "navigation criterion" through the "compass" icon located at the top right of each application.

Figure 3.24 shows the distribution of the answers to each positive tone (top) and negative tone (bottom) questions. The appreciation appears generally high for the search section, for the presence of a contextual list of the apps and of the forward / backward navigation cursors (questions NR1-NR3; on average: 83.5% positive, 12.5% neutral and 4% negative responses). The effectiveness of the navigation mechanism for the single app gathered slightly lower appreciation (question NR4: 74% positive, 19% neutral and 8% negative) probably due to its particularity and the uncommon feature it offers.

User contributions. The last seven questions investigated about the quality of the features, introduced for giving a score to each app, and the importance to leave comments, highlight inaccuracies and share use cases:

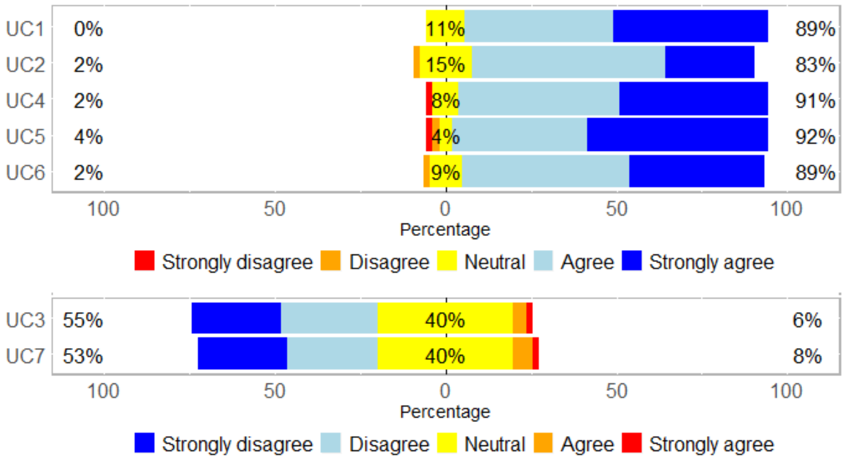

Figure 3.25: The distributions of the answers to positive tone (top) and negative tone (bottom) questions relative to the User Contribution (UC) features. In the second plot, the color scale has been reversed to map, as in the first plot, positive values to azure/blue colors.

UC1 I consider useful for users to evaluate applications;

UC2 I find clear and understandable the five evaluation items (functionality, ease to use, applicability, originality, overall opinion);

UC3 I would have expected other evaluation parameters or changed existing ones;

UC4 I find useful for users to comment on applications;

UC5 I find important to have the opportunity to suggest changes and communicate inaccuracies in the information cards;

UC6 I consider significant to be able to share use cases of the application;

UC7 I would have preferred to login on the platform to post comments.

Figure 3.25 shows the distribution of the answers to positive tone (top) and negative tone (bottom) questions. The possibility of evaluating the apps has been generally considered useful and the evaluation items understandable (questions UC1, UC2: 86% positive, 13% neutral and 1% negative) while there is a considerable uncertainty about the choice of such evaluation parameters (question UC3: 55% positive, 40% neutral, 6% negative). Also, the possibility to comment the apps, suggest changes, suggest inaccuracies and share use cases

(questions UC4-UC6) are valued positively from about 90% of the sample. Anonymous comments are approved from 53% of the sample (question UC7) with a significant part of users neutral about this choice.
Overall, the results of the user evaluation encourage us to continue experimenting with and improving the model in addition to explore new approaches.

3.8.4 The comparative study

A second study was organized as a multi-factor within-subject study in order to collect comparative user opinions about AppInventory and two other Web catalog of applications: Edshelf [11] and Essediquadro [14]. In addition to the SUS related to the three considered platforms, the study acquired 22 questions on four aspects of the platform: user layout, semantic structure, navigation/research mechanisms and user contribution features plus an additional question about the overall system.

Participants The study involved 31 persons (28 F, 3 M) of age between 20 and 49 years, with a mean age of 24.9 and deviation standard 5.3; most of them were students (84%) attending a course of Web technologies in University of Udine.

Procedure and results Before conducting the studio, the three platforms were presented to the participants, illustrating in details the specific features and proposing some tasks to familiarize with them: follow the guided tour, where available; explore some of the available categories; analyze the information cards of some applications and the related comment / evaluation sections; carry out a simple and advanced search and browse the results; apply different sorting criteria. The questionnaire was organized in sections in order to collect user opinions about the four aspects already considered in the preliminary study: User layout (UL), Semantic structure (SS), Navigation and search features (NR) and user contributions (UC). Next a SUS questionnaire was proposed for each platform and finally users were asked to assign an overall score to each platform.

Metrics	AppInventory	Edshelf	Essediquadro
Sample size	31	31	31
Min	48.00	15.00	12.00
1st Qu.	72.00	41.00	32.00
Median	80.00	58.00	42.00
Mean	79.29	57.52	43.39
Std. dev.	13.16	20.04	17.95
3rd Qu.	89.00	73.50	58.50
Max	100.00	90.00	72.00

Table 3.2: The SUS distributions of the three platforms.

Table 3.2 and Figure 3.26 describe the distribution of the SUS for each platform. The results confirm very similar values of SUS for AppInventory in this and previous study (the differences between mean, median, 1st and 3rd quartile of the two distributions are lower than 2 units). In order to compare the results we applied a hypothesis test for the difference between m_a and m_e (the AppInventory and Edshelf SUS medians) and m_e and m_s (the Edshelf and Essediquadro SUS medians), fixing the null hypotheses $H0_{ae}$:

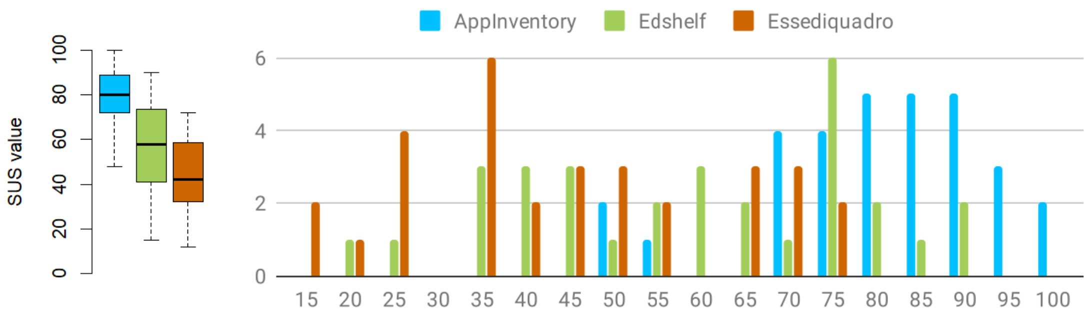

Figure 3.26: The comparison of the SUS distributions of the three platforms (left) and the absolute frequencies on 5-units intervals.

$m_a = m_e$ and $H0_{es}$: $m_e = m_s$. Applying a Wilcoxon signed-ranks test we get $W_{ae} = 39$ and $W_{es} = 41.5$ which are below the respective critical values 120 and 92 for $p < .01$, leading us to reject both $H0_{ae}$ and $H0_{es}$ and assert the significance of the differences of medians. Figure 3.27 shows the comparison of the answers to the single questions of the SUS questionnaire.

In order to investigate the four aspects (UL, SS, NR, UC) considered in the first study, we reformulated the questions in a more general form to make them applicable to the all considered platforms. About the *User Layout* (UL) we asked users how much do they agree with each of the following statements:

UL1-c I believe that the user interface adopted for the main page of the catalog is innovative and intuitive;

UL2-c I find that the main page provides an effective overview of the catalog;

UL3-c I find that the graphic elements (icons, titles, sections, ...) are understandable;

UL4-c I find it easy to identify the number of applications in the category;

UL5-c I believe that the platform offers effective tools to learn using it (eg. Quick start guides, guided tours, contextual help, FAQ and support pages, ...).

The Figure 3.28 summarizes the results: the positive responses vary from 87% and 97% for AppInventory, from 19% to 48% for Edshelf and from 3% and 26% for Essediquadro.

About the *Semantic Structure* the six sentences we asked users to evaluate for the three platforms were:

SS1-c I think the proposed classification helps me to orient myself between the apps;

SS2-c I think the number of categories provided is excessive or insufficient;

SS3-c I consider the information cards of the single applications complete and well detailed also for the presence of multimedia contents;

SS4-c I think that the information card does not contain important information regarding the application;

SS5-c Looking at the card, I can get a general and complete idea about the application;

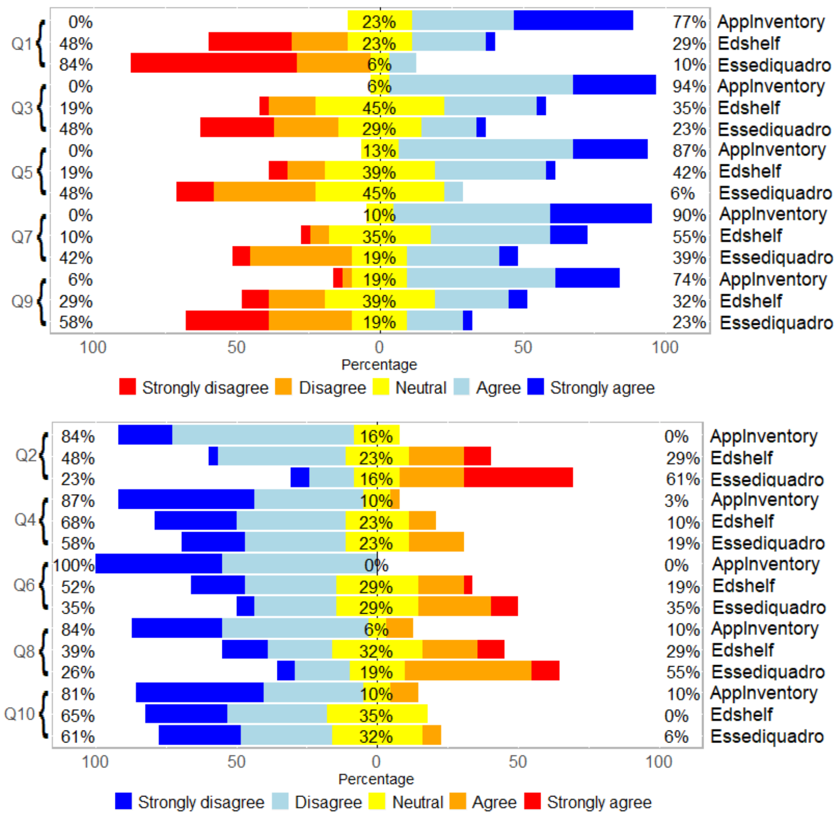

Figure 3.27: The distributions of the answers to the odd, positive tone, SUS questions (top) and to the even, negative tone, ones (bottom) for the 3 platforms. In the second plot, the color scale has been reversed to map, as in the first plot, positive values to azure/sky colors.

SS6-c I believe that the card of an app provides me with comprehensive information on the classification of the application (eg. if the app is present in more than one category and with what degree, the associated tags, additional classification taxonomies, etc.).

The results are presented in Figure 3.29, separately for the positive and negative tone questions. For this set of questions AppInventory collected positive answers in a percentage between 84% and 97%, Edshelf between 35% and 71%, Essediquadro between 13% and 58%.

The *Navigation and Research mechanisms* were investigated through the following six questions:

NR1-c I think the search functions are easily identifiable and usable;

NR2-c I found the basic and advanced search understandable and functional;

NR3-c I find incomplete the search filters;

NR4-c I consider appropriate and complete the sorting criteria of the app;

NR5-c The platform helps me find similar apps by letting me choose the similarity criteria;

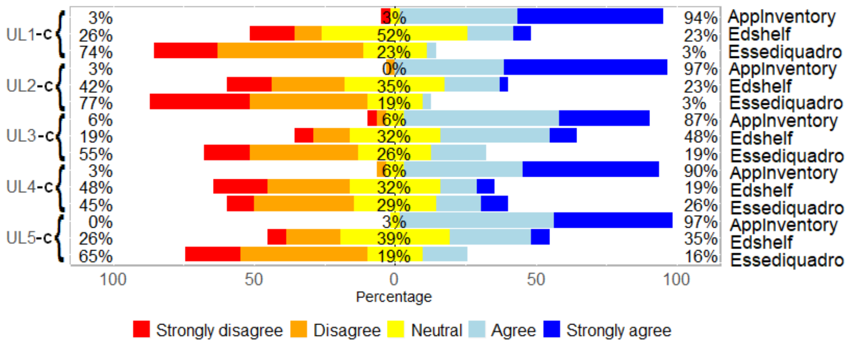

Figure 3.28: The distributions of the answers to (all positive tone) questions relative to the User Layout (UL) aspects for the three platforms.

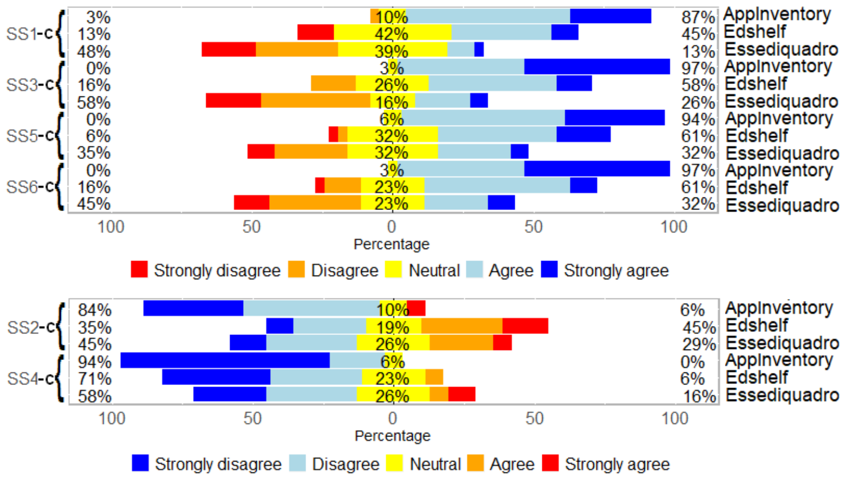

Figure 3.29: The distributions of the answers to positive tone (top) and negative tone (bottom) questions relative to the Semantic Structure (SS) aspects for the three platforms. In the second plot, the color scale has been reversed to map, as in the first plot, positive values to azure/sky colors.

NR6-c The platform supports me in finding applications for a certain goal.

The results, illustrated in Figure 3.30, show a percentage of positive answers for AppInventory between 87% and 100%, for EdShelf between 35% and 65% while for Essediquadro between 16% and 32%.

The last investigated aspect was about *User Contributions* features:

UC1-c I believe that user contributions are important in a catalog of applications;

UC2-c I believe that the platform provides good support for users to add comments on applications;

UC3-c I would have expected more evaluation parameters or change existing ones;

UC4-c I believe that the platform provides users with good support for suggesting integrations and reporting inaccuracies in the descriptions;

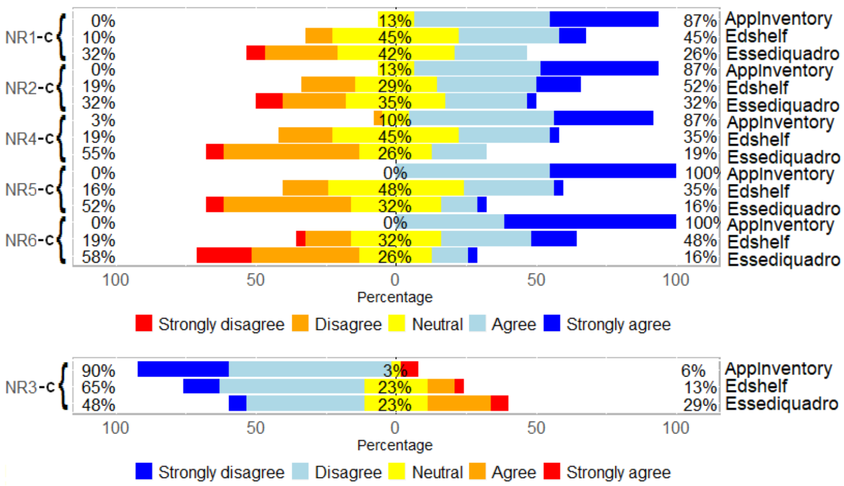

Figure 3.30: The distributions of the answers to positive tone (top) and negative tone (bottom) questions relative to the Navigation and Research mechanisms (NR) aspects for the three platforms. In the second plot, the color scale has been reversed to map, as in the first plot, positive values to azure/sky colors.

UC5-c I believe that the platform provides users with good support for sharing educational use-cases of the applications.

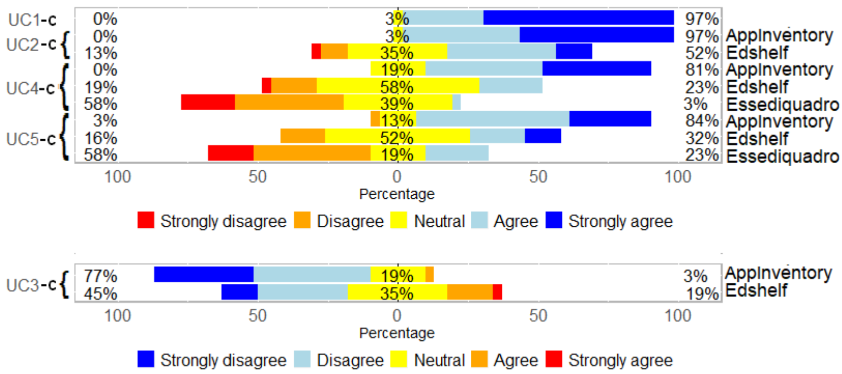

Figure 3.31: The distributions of the answers to positive tone (top) and negative tone (bottom) questions relative to the User Contributions (UC) features for the three platforms. In the second plot, the color scale has been reversed to map, as in the first plot, positive values to azure/sky colors.

The first question was general and not referred to any platform, the questions UC2-c and UC3-c apply only to AppInventory and Edshelf since Essediquadro does not accept user comments and ratings. The results are presented in Figure 3.31.

In the last question (G1-c) we asked users to formulate an overall rating, in a scale from *1=very bad* to *5=very good* of the three platforms. Figure 3.32 shows the results which are very positive for AppInventory (94% of sample attributed a score greater or equals to 4), Edshelf is positively evaluated by the 32% of the sample with a large percentage of neutral scores while Essediquadro gets only 3% of positive scores.

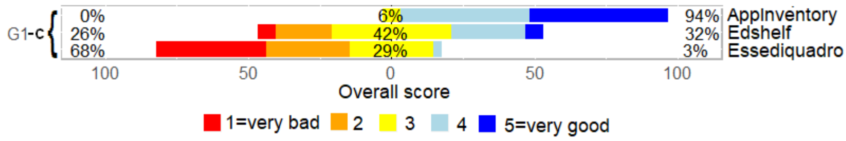

Figure 3.32: The distributions of the user overall ratings for the three platforms.

A comparison between the SUS medians (Table 3.3) of the three platforms and the weighted average of the overall ratings normalized on a scale from *1 to 100* shows how the differences in the perceived usability, already revealed by the SUS, are more marked when we consider the overall features of the three platforms.

Metrics	AppInventory	Edshelf	Essediquadro
SUS Median	80.00	58.00	42.00
Overall rating	85.48	51.61	24.19

Table 3.3: The comparison between the SUS medians and the normalized overall ratings of the three platforms.

3.9 Conclusion and future work

In this chapter we presented AppInventory, a Web platform designed to allow teachers to browse a repository of applications, organized in a purpose-based taxonomy, using a visual approach. It offers a novel modality for representing and exploring the catalog. Two usability evaluation were performed applying qualitative and comparative tests. They were largely discussed in Section 3.8: the results are encouraging and highlighted the positive impact of our visual model, with respect to the general platform and also specifically to the four considered features. The final comparative study shows that AppInventory totals at least 30% more positive reviews by users than those of Edshelf and Essediquadro, on all analyzed aspects.

The AppInventory platform represents an original contribution of this thesis both for the semantic model based on the zz-structures, the newly zz-view proposed and implemented, the navigation solutions introduced, based on semantic zoom and contextual browsing of the applications, the collecting of user contributioons and, finally, the visual comparison of app metadata.

The limitations of the proposed approach mainly regards the fruition of the catalog from mobile devices: although possible, the navigation is rather slowed down due to the size of the DOM and the use of SVG elements that require significant resources for their processing also disabling some animations, active in the desktop version. A possible approach totally based on HTML5 standard will be investigated in order to get a better user experience on mobile devices, improving the responsiveness of the platform.

Future work will also involve the constant updating of the information published in the catalog; the development of new views and features and the experimentation of a recommender system based on zz-structures.

Conclusions

In this thesis, we investigated possible visual approaches for the representation of the zz-structures. Starting from its formal description and model, we proposed new zz-views and applied them in the context of two different case studies where we combined holistic views with a contextual and multilevel exploration mechanism. We also presented some guidelines for the design of visualization systems, some classifications of visual representations of knowledge and possible technologies / frameworks for their implementation in the context of Web applications.

Before analyzing, modeling and developing the two case studies, we implemented a first prototype for experimenting the effectiveness of the newly introduced proposals of deep and narrative zz-views, dedicated to the representation of a scientific bibliography. The aim was to realize a Web application for exploring, in a visual and original way, the whole collection of scientific researches about zz-structures. The narrative-view, enriched by the user interactions management, appeared to be an interesting solution for the representation of: the relationships between authors and papers, the collaboration network, and the temporal collocations of the papers. At the same time, the deep-view enables users to display an overlay with the citation network of a given paper.

From this first prototype, we conceived the idea of applying this approach to any bibliography. We started investigating the possibility of obtaining, in real-time, metadata of papers and authors by querying one or more bibliographic indexes, such as Scopus, Web of Science, etc. From a detailed analysis of the exposed metadata and the API services offered by a series of bibliographic indexes, we started to implement a Web application, we called VisualBib, for supporting researchers in the progressive building, enriching and sharing scientific bibliographies. VisualBib evolved over time from version 1.0 released on February 2018, characterized by a data retrieval limited to two sources (Scopus and OpenCitations) and a basic metadata management, to version 2.0, released in September of the same year, which introduced some new features: import/export from/to BibTeX archives, new sources of metadata (CrossRef and Orcid), merge of duplicated author names, bibliography sharing, and others.

The evaluation studies carried out on this version highlighted the positive impact of our visual model and encouraged us to improve it and develop further visual analysis features we incorporated in the version 3.0 of the application, released in October 2019.

This last version introduced several new features, like the retrieval and management of extended metadata for papers and authors, an extension of the narrative views to subject areas, keywords and tags and a completely renovated interface which enables effective visual and quantitative analysis of bibliographies. A user study stated a good appreciation of the various new features (from 72% to 84% of positive scores) and no significant decrease of usability, compared to the version 2.0.

VisualBib, whose name has been registered by University of Udine, is currently a “live system”, available online for research and personal purposes (not commercial ones) and represents one of the few solutions of visual analysis environment for bibliographies working on real-time data and not on static datasets.

The system is open to future improvements, such as interfacing with further bibliographic indexes, enhancing of the scalability of the system and incorporating of new features.

The second case study concerned the modeling and development of a multimedia catalog of Web and mobile applications. We analyzed and documented 281 applications, preparing for each of them a detailed multilingual card and a video-presentation, organizing all the material in an original purpose-based taxonomy, visually represented through a browsable holistic view. The catalog, we called AppInventory, offers an innovative approach for its exploration, both through a continuous semantic zoom to progressively discover details initially hidden and a contextual exploration mechanisms based on zz-structures. AppInventory is available online since November 2018 and has been recently improved by adding new features such as a visual analysis tools for the comparison of the applications data and an interactive guided tour of the platform. AppInventory also collects user contributions and evaluations about the apps that can help users to select them.

The results of two user studies carried out on groups of teachers and students shown a very positive impact of our proposal in term of graphical layout, semantic structure, navigation mechanisms and usability, also in comparison with two similar catalogs.

In summary, the original contributions of this thesis are:

1. the extending of the existing models of zz-views with three new proposals: *deep-views*, *narrative-views* and *bubble-views*;

2. the publication of an interactive visual representation of a comprehensive bibliography about zz-structures, accessible at `http://zzstructure.uniud.it`;

3. the modeling and realization of the VisualBib Web platform for supporting researchers in the building, refining, representing, analyzing and sharing of scientific bibliographies, accessible at `http://visualbib.uniud.it`;

4. the modeling and realization of AppInventory, a multimedia catalog of 281 Web 2.0 and mobile applications, accessible at `http://appinventory.uniud.it`, based on an original purpose-based taxonomy, for supporting teachers and students in selecting best apps for learning activities.

One of my personal goals during this Ph.D. course was to be able to reconcile research with the concrete development of free tools that could be useful to teachers and researchers; I hope I did, at least in part.

A
Appendices

A.1 Sample bibliography for evaluation

Sample bibliography in BibTeX format used in the evaluation of VisualBib 3.0 platform discussed in Subsection 2.9.2 and in the use case scenario of Section 2.6.1. This bibliography[1], imported in the system, will be automatically enriched through the *seeking metadata* function. For this reason most of the references contains only basic metadata: title, year and a univocal identificator (doi or scopusid). The presence of this identifier enables VisualBib to retrieve extended metadata from Scopus APIs including the list of authors, the source, the publisher, the subject area, the keywords, the abstract, etc. Appendix A.2 lists the enriched version of this bibliography after the merging of duplicated authors and the integration of additional three papers.

```
@inproceedings{Dattolo2009ASA,
    title = {A State of Art Survey on zz-structures},
    year = {2009},
    scopusid={84891551906}
}
@Inproceedings{Dattolo2009AFD,
    title = {A formal description of zz-structures},
    year = {2009},
    scopusid={84891531311}
}
@inproceedings{Bergstrom2009,
    title={Augmenting the exploration of digital libraries with web-based visualizations},
    year={2009},
    doi={10.1109/ICDIM.2009.5356798}
}
@article{Eck2010,
    title={Software survey: VOSviewer, a computer program for bibliometric mapping.},
    year={2010},
    scopusid={77953711904}
}
@article{Chen2010,
    author={Chen, C. and Ibek San Juan, F. and Hou, J.},
    title={The structure and dynamics of co-citation clusters: A multiple-perspective
    co-citation analysis},
    year={2010},
    scopusid={77954068456}
}
@inproceedings{Costagliola2011,
    title={CyBiS: A Novel Interface for Searching Scientific Documents},
    year={2011},
    doi={10.1109/IV.2011.95},
    scopusid={80052986332}
}
@article{Dork2012,
    title={Pivotpaths: Strolling through faceted information spaces},
    year={2012},
    scopusid={84867642651}
}
@inproceedings{Matejka2012,
    title = {Citeology: Visualizing Paper Genealogy},
```

[1] Accessible online at `http://visualbib.uniud.it/biblio-example-in.bib`

```
    year = {2012},
    scopusid={84862663957}
}
@article{brooke2013,
    title={SUS: a retrospective},
    author={Brooke, John},
    journal={Journal of usability studies},
    volume={8},
    number={2},
    pages={29--40},
    year={2013},
    publisher={Usability Professionals' Association}
}
@article{Kihm2013,
    title={Combining Computational Analyses and Interactive Visualization for
    Document Exploration and Sensemaking in Jigsaw},
    year={2013},
    scopusid={84883074568}
}
@article{vanEck2014,
    title = {CitNetExplorer: A new software tool for analyzing and visualizing
    citation networks},
    year = {2014},
    doi = {10.1016/j.joi.2014.07.006}
}
@inproceedings{Kucher2015,
    title={Text visualization techniques: Taxonomy, visual survey, and community insights},
    year={2015},
    scopusid={84942235504}
}
@article{Federico2017,
    title={A Survey on Visual Approaches for Analyzing Scientific Literature and Patents},
    year={2017},
    doi={10.1109/TVCG.2016.2610422},
}
@inbook{Corbatto2018vb,
    author={Dattolo, Antonina and Corbatto, Marco},
    title={A Web application for creating and sharing visual bibliographies},
    year={2018},
    scopusid={85058996659}
}
@inproceedings{Dattolo2018Vis,
    author={Dattolo, Antonina and Corbatto, Marco},
    title={VisualBib: Narrative Views for Customized Bibliographies},
    year={2018},
    doi={10.1109/iV.2018.00033}
}
```

A.2 BibTeX of the enriched bibliography for evaluation

Below is the sample bibliography of Appendix A.1 after the application of *seek metadata* and *author matching* functions, the addition of three new papers and the exporting in BibTeX format, as discussed in the use case scenario in Subsection 2.9.2. In addition to several metadata fields, the output archive contains further fields, needed to reconstruct, in a next importing into VisualBib, the exact authors/papers relationships and the citation network:

- *authordata* which uniquely indexes the authors of each paper by Orcid or Scopus author id;
- *references* which indexes the cited papers inside the bibliography;

- *keywords* and *author_keywords*: respectively the list of keywords attributed by the system (Scopus) and those attributed by the authors.

A direct reference to this bibliography is: `http://bit.ly/vb3-evBib`.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bibtex Exported from VisualBib
% http://visualBib.uniud.it/
%
% Created on: Oct 22, 2019
%
% Project_Title: Evaluation VisualBib - example
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
@inproceedings{84891551906,
 author={Dattolo Antonina and Luccio F.},
 title={A state of art survey on zz-structures},
 authordata={O:0000-0002-8511-524X|S:6602802183,O:0000-0002-5409-5039|S:7005244352},
 booktitle={CEUR Workshop Proceedings},
 references={84891531311},
 scopusid={84891551906},
 url={https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=84891551906},
 month={12},
 pages={1-6},
 volume={508},
 issn={16130073},
 citedby={11},
 author_keywords={},
 keywords={Formal Description;State of the art},
 topics={Computer Science (all)},
 year={2009},
 source={Scopus}
}
@inproceedings{84891531311,
 author={Dattolo Antonina and Luccio F.},
 title={A formal description of zz-structures},
 authordata={O:0000-0002-8511-524X|S:6602802183,O:0000-0002-5409-5039|S:7005244352},
 booktitle={CEUR Workshop Proceedings},
 scopusid={84891531311},
 url={https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=84891531311},
 month={12},
 pages={7-11},
 volume={508},
 issn={16130073},
 citedby={12},
 author_keywords={},
 keywords={Formal Description;Innovative structures},
 topics={Computer Science (all)},
 year={2009},
 source={Scopus}
}
@inproceedings{10.1109/ICDIM.2009.5356798,
 author={Bergström P. and Atkinson D.},
 title={Augmenting the exploration of digital libraries with web-based visualizations},
 authordata={S:57196791745,S:7202372426},
 booktitle={4th International Conference on Digital Information Management, ICDIM 2009},
 doi={10.1109/ICDIM.2009.5356798},
 scopusid={76249098696},
 url={https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=76249098696},
 month={12},
 pages={63-69},
 isbn={9781424442539},
 citedby={17},
 author_keywords={},
 keywords={Citation networks;Cognitive loads;Research papers;User study;
 Web-based applications;Web-based visualization;Web-page},
 topics={Computer Science (all)},
 year={2009},
 source={Scopus}
}
@article{10.1007/s11192-009-0146-3,
 author={Van Eck N. and Waltman L.},
 title={Software survey: VOSviewer, a computer program for bibliometric mapping},
 authordata={O:0000-0001-8448-4521|S:14632651000,S:14632830700},
```

```
 journal={Scientometrics},
 doi={10.1007/s11192-009-0146-3},
 scopusid={77953711904},
 url={https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=77953711904},
 month={1},
 publisher={Springer Netherlands},
 pages={523-538},
 volume={84},
 issn={01389130},
 citedby={1004},
 author_keywords={Bibliometric mapping;Journal co-citation analysis;
 Science mapping;Visualization;VOS;VOSviewer},
 keywords={},
 topics={Social Sciences (all);Computer Science Applications;Library and Information Sciences},
 year={2010},
 source={Scopus}
}
@article{10.1002/asi.21309,
 author={Chen C. and Ibekwe-SanJuan F. and Hou J.},
 title={The structure and dynamics of cocitation clusters: A multiple-perspective
 cocitation analysis},
 authordata={O:0000-0001-8584-1041|S:7501950297,S:56192063200,S:56648784400},
 journal={Journal of the American Society for Information Science and Technology},
 doi={10.1002/asi.21309},
 scopusid={77954068456},
 url={https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=77954068456},
 month={7},
 pages={1386-1409},
 volume={61},
 issn={15322882 15322890},
 citedby={282},
 author_keywords={},
 keywords={Analysis method;Author cocitation analysis;Cluster labeling;Co-citation networks;
 Cocitation;Generic method;Integrating networks;Interpretability;Sense making;
 Spectral clustering;Structure and dynamics;Text summarization},
 topics={Software;Information Systems;Human-Computer Interaction;
 Computer Networks and Communications;Artificial Intelligence},
 year={2010},
 source={Scopus}
}
@inproceedings{10.1109/IV.2011.95,
 author={Costagliola G. and Fuccella V.},
 title={CyBiS: A novel interface for searching scientific documents},
 authordata={O:0000-0003-3816-7765|S:7003667801,S:9640308300},
 booktitle={Proceedings of the International Conference on Information Visualisation},
 references={10.1109/ICDIM.2009.5356798},
 doi={10.1109/IV.2011.95},
 scopusid={80052986332},
 url={https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=80052986332},
 month={9},
 pages={276-281},
 isbn={9780769544762},
 issn={10939547},
 citedby={6},
 author_keywords={3D;CyBiS;Cylindrical Biplot System;paper search;visualization},
 keywords={3D;Collection of documents;CyBiS;Cylindrical Biplot System;
 Scientific documents;Scientific papers;Visualization tools},
 topics={Software;Signal Processing;Computer Vision and Pattern Recognition},
 year={2011},
 source={Scopus}
}
@article{10.1109/TVCG.2012.252,
 author={Dörk M. and Henry-Riche N. and Ramos G. and Dumais S.},
 title={Pivot paths: Strolling through faceted information spaces},
 authordata={S:23134745400,S:35754698000,S:55993680200,S:7003862762},
 journal={IEEE Transactions on Visualization and Computer Graphics},
 references={10.1145/2212776.2212796},
 doi={10.1109/TVCG.2012.252},
 scopusid={84867642651},
 url={https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=84867642651},
 month={10},
 pages={2709-2718},
 volume={18},
```

```
 issn={10772626},
 citedby={98},
 author_keywords={animation;exploratory search;information seeking;Information visualization;
 interactivity;node-link diagrams},
 keywords={Exploratory search;Information seeking;Information visualization;
 Interactivity;Node-link diagrams},
 topics={Software;Signal Processing;Computer Vision and Pattern Recognition;
 Computer Graphics and Computer-Aided Design},
 year={2012},
 source={Scopus}
}
@inproceedings{10.1145/2212776.2212796,
 author={Matejka J. and Grossman T. and Fitzmaurice G.},
 title={Citeology: Visualizing paper genealogy},
 authordata={S:10044259500,S:7003520062,S:7005241818},
 booktitle={Conference on Human Factors in Computing Systems - Proceedings},
 doi={10.1145/2212776.2212796},
 scopusid={84862663957},
 url={https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=84862663957},
 month={6},
 pages={181-189},
 isbn={9781450310161},
 citedby={37},
 author_keywords={citations;information visualization;references},
 keywords={citations;Family tree;Information visualization;
 Interactive visualizations;references},
 topics={Software;Human-Computer Interaction;Computer Graphics and Computer-Aided Design},
 year={2012},
 source={Scopus}
}
@article{0,
 author={Brooke John},
 title={SUS: a retrospective},
 authordata={I:3},
 journal={Journal of usability studies},
 publisher={Usability Professionals’ Association},
 pages={29-40},
 volume={8},
 author_keywords={},
 keywords={},
 topics={},
 tags={},
 year={2013},
 source={External}
}
@article{10.1109/TVCG.2012.324,
 author={Görg C. and Liu Z. and Kihm J. and Choo J. and Park H. and Staško J.},
 title={Combining computational analyses and interactive visualization for
 document exploration and sensemaking in jigsaw},
 authordata={S:56124473700,S:55714445500,S:36727682900,S:24512223400,S:7601569954,
 S:7006755495},
 journal={IEEE Transactions on Visualization and Computer Graphics},
 doi={10.1109/TVCG.2012.324},
 scopusid={84883074568},
 url={https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=84883074568},
 month={9},
 pages={1646-1663},
 volume={19},
 issn={10772626},
 citedby={49},
 author_keywords={document analysis;exploratory search;information seeking;
 information visualization;sensemaking;Visual analytics},
 keywords={Document analysis;Exploratory search;Information seeking;
 Information visualization;Sensemaking;Visual analytics},
 topics={Software;Signal Processing;Computer Vision and Pattern Recognition;
 Computer Graphics and Computer-Aided Design},
 year={2013},
 source={Scopus}
}
@article{10.1016/j.joi.2014.07.006,
 author={Van Eck N. and Waltman L.},
 title={CitNetExplorer: A new software tool for analyzing and visualizing citation networks},
 authordata={O:0000-0001-8448-4521|S:14632651000,S:14632830700},
```

journal={Journal of Informetrics},
references={10.1007/s11192-009-0146-3},
doi={10.1016/j.joi.2014.07.006},
scopusid={84906491664},
url={https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=84906491664},
month={1},
publisher={Elsevier Ltd},
pages={802-823},
volume={8},
issn={17511577},
citedby={89},
author_keywords={Citation network;CitNetExplorer;Computer software;
Network analysis;Visualization},
keywords={Citation networks;CitNetExplorer;Community detection;
Research fields;Research topics;Scientific publications;Scientometrics;
Technical details;Citnetexplorer},
topics={Computer Science Applications;Library and Information Sciences},
year={2014},
source={Scopus}
}
@inproceedings{10.1109/PACIFICVIS.2015.7156366,
author={Kucher K. and Kerren A.},
title={Text visualization techniques: Taxonomy, visual survey, and community insights},
authordata={O:0000-0002-1907-7820|S:56647973300,O:0000-0002-0519-2537|S:6508221631},
booktitle={IEEE Pacific Visualization Symposium},
doi={10.1109/PACIFICVIS.2015.7156366},
scopusid={84942235504},
url={https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=84942235504},
month={1},
publisher={IEEE Computer Society help@computer.org},
pages={117-121},
volume={2015-July},
isbn={9781467368797},
issn={21658773 21658765},
citedby={54},
author_keywords={community analysis;interaction;survey;taxonomy;text visualization;
Visualization;web-based systems},
keywords={Community analysis;Gaining insights;Information visualization;interaction;
Research trends;Text visualization;Visual metaphor;Web-based system},
topics={Computer Graphics and Computer-Aided Design;Computer Vision and Pattern Recognition;
Hardware and Architecture;Software},
year={2015},
source={Scopus}
}
@article{10.1109/TVCG.2016.2610422,
author={Federico P. and Heimerl F. and Koch S. and Miksch S.},
title={A Survey on Visual Approaches for Analyzing Scientific Literature and Patents},
authordata={O:0000-0002-1830-0330|S:52563493000,S:12781821100,S:24070759300,S:7003690425},
journal={IEEE Transactions on Visualization and Computer Graphics},
references={10.1109/PACIFICVIS.2015.7156366,10.1109/IV.2011.95,10.1109/TVCG.2012.324,
10.1109/ICDIM.2009.5356798,10.1016/j.joi.2014.07.006,10.1007/s11192-009-0146-3,
10.1145/2212776.2212796,10.1109/TVCG.2012.252,10.1109/ICDIM.2009.5356798,
10.1007/s11192-009-0146-3,10.1109/IV.2011.95,10.1145/2212776.2212796,10.1109/TVCG.2012.252,
10.1109/TVCG.2012.324,10.1016/j.joi.2014.07.006,10.1109/PACIFICVIS.2015.7156366,
10.1109/ICDIM.2009.5356798,10.1007/s11192-009-0146-3,10.1109/IV.2011.95,
10.1145/2212776.2212796,10.1109/TVCG.2012.252,
10.1109/TVCG.2012.324,10.1016/j.joi.2014.07.006,10.1109/PACIFICVIS.2015.7156366,
10.1109/TVCG.2015.2467621},
doi={10.1109/TVCG.2016.2610422},
scopusid={85027016560},
url={https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=85027016560},
month={1},
publisher={IEEE Computer Societyhelp@computer.org},
volume={PP},
issn={10772626},
citedby={19},
author_keywords={Documents;Patents;Scientific Literature;Survey;Visualization},
keywords={Application scenario;Documents;Efficient analysis;Interactive analysis;
Patents;Scientific articles;Scientific literature;Scientific progress;
Bottom up approach;documents;patents},
topics={Software;Signal Processing;Computer Vision and Pattern Recognition;
Computer Graphics and Computer-Aided Design},
year={2016},

```
 source={Scopus}
}
@book{10.1007/978-3-030-01379-0_6,
 author={Corbatto Marco and Dattolo Antonina},
 title={A Web Application for Creating and Sharing Visual Bibliographies},
 authordata={O:0000-0003-0723-8241|S:6507516644,O:0000-0002-8511-524X|S:6602802183},
 booktitle={Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial
 Intelligence and Lecture Notes in Bioinformatics)},
 references={10.1109/TVCG.2016.2610422,10.1109/TVCG.2012.324,10.1145/2212776.2212796,
 10.1109/PACIFICVIS.2015.7156366,10.1109/TVCG.2012.252,10.1016/j.joi.2014.07.006,
 10.1109/ICDIM.2009.5356798,84891531311,84891551906,10.1109/TVCG.2016.2610422,
 10.1109/TVCG.2016.2610422},
 doi={10.1007/978-3-030-01379-0_6},
 scopusid={85058996659},
 url={https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=85058996659},
 month={1},
 publisher={Springer Verlagservice@springer.de},
 pages={78-94},
 volume={10959 LNCS},
 isbn={9783030013783},
 issn={16113349 03029743},
 citedby={4},
 author_keywords={Graphic organizer;Holistic view;Human computer interaction (HCI);
 Information visualization;Visual bibliography;Visualization design and evaluation methods},
 keywords={Graphic organizers;Holistic view;Human Computer Interaction (HCI);
 Information visualization;Visualization designs},
 topics={Theoretical Computer Science;Computer Science (all)},
 year={2018},
 source={Scopus}
}
@inproceedings{10.1109/iV.2018.00033,
 author={Dattolo Antonina and Corbatto Marco},
 title={VisualBib: Narrative views for customized bibliographies},
 authordata={O:0000-0002-8511-524X|S:6602802183,O:0000-0003-0723-8241|S:6507516644},
 booktitle={Information Visualisation - Biomedical Visualization,
 Visualisation on Built and Rural Environments and Geometric Modelling and Imaging, IV 2018},
 references={10.1109/TVCG.2016.2610422,10.1109/PACIFICVIS.2015.7156366,
 10.1109/ICDIM.2009.5356798,10.1109/IV.2011.95,10.1145/2212776.2212796,
 10.1109/TVCG.2012.252,10.1002/asi.21309,10.1109/TVCG.2012.324,
 10.1016/j.joi.2014.07.006,84891551906,84891531311,10.1007/978-3-030-01379-0_6,
 10.1109/TVCG.2016.2610422,10.1109/TVCG.2016.2610422},
 doi={10.1109/iV.2018.00033},
 scopusid={85059038899},
 url={https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=85059038899},
 month={12},
 publisher={Institute of Electrical and Electronics Engineers Inc.},
 pages={133-138},
 isbn={9781538672020},
 citedby={2},
 author_keywords={Graphic organizer;Holistic view;Narrative view;Visual bibliography},
 keywords={Citation indexes;Explicit representation;Graphic organizers;Holistic view;
 Narrative view;Scientific literature;Visual representations;WEB application},
 topics={Computer Graphics and Computer-Aided Design;Information Systems;
 Media Technology;Modeling and Simulation},
 year={2018},
 source={Scopus}
}
@inproceedings{10.1145/3231622.3232505,
 author={Kucher K. and Skeppstedt M. and Kerren A.},
 title={Application of interactive computer-assisted argument
 extraction to opinionated social media texts},
 authordata={O:0000-0002-1907-7820|S:56647973300,S:55605433600,
 O:0000-0002-0519-2537|S:6508221631},
 booktitle={ACM International Conference Proceeding Series},
 references={10.1109/TVCG.2012.324},
 doi={10.1145/3231622.3232505},
 scopusid={85055567433},
 url={https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=85055567433},
 month={8},
 publisher={Association for Computing Machineryacmhelp@acm.org},
 pages={102-103},
 isbn={9781450365017},
 citedby={0},
```

```
 author_keywords={Annotation;Argument extraction;Interaction;Sentiment analysis;
 Sentiment visualization;Stance analysis;Stance visualization;Text visualization;
 Topic modeling;Visualization},
 keywords={Annotation;Interaction;Stance analysis;Text visualization;Topic Modeling},
 topics={Software;Human-Computer Interaction;Computer Vision and Pattern Recognition;
 Computer Networks and Communications},
 year={2018},
 source={Scopus}
}
@article{10.1109/TVCG.2015.2467621,
 author={Heimerl F. and Han Q. and Koch S. and Ertl T.},
 title={CiteRivers: Visual Analytics of Citation Patterns},
 authordata={S:12781821100,S:56437203600,S:24070759300,S:7004765655},
 journal={IEEE Transactions on Visualization and Computer Graphics},
 references={10.1109/TVCG.2012.252,10.1109/TVCG.2012.324},
 doi={10.1109/TVCG.2015.2467621},
 scopusid={84946605102},
 url={https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=84946605102},
 month={1},
 publisher={IEEE Computer Society help@computer.org},
 pages={190-199},
 volume={22},
 issn={10772626},
 citedby={35},
 author_keywords={Color;Data mining;Joining processes;Market research;Metadata;
 Tag clouds;Visualization},
 keywords={Document metadatas;Interactive analysis;Interactive visualizations;Joining process;
 Long-term development;Market researches;Scientific literature;Tag clouds},
 topics={Software;Signal Processing;Computer Vision and Pattern Recognition;
 Computer Graphics and Computer-Aided Design},
 year={2016},
 source={Scopus}
}
@article{10.1016/j.knosys.2019.07.031,
 author={Dattolo Antonina and Corbatto Marco},
 title={VisualBib: A novel Web app for supporting researchers in the creation, visualization
 and sharing of bibliographies},
 authordata={O:0000-0002-8511-524X|S:6602802183,O:0000-0003-0723-8241|S:6507516644},
 journal={Knowledge-Based Systems},
 references={10.1109/TVCG.2016.2610422,10.1007/978-3-030-01379-0_6,10.1109/iV.2018.00033,
 10.1109/PACIFICVIS.2015.7156366,10.1002/asi.21309,10.1007/s11192-009-0146-3,
 10.1109/IV.2011.95,10.1145/2212776.2212796,10.1109/TVCG.2012.252,
 10.1016/j.joi.2014.07.006,84891531311,84891551906},
 doi={10.1016/j.knosys.2019.07.031},
 scopusid={85069825956},
 url={https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=85069825956},
 month={10},
 publisher={Elsevier B.V.},
 volume={182},
 issn={09507051},
 citedby={0},
 author_keywords={Bibliographic indexes;Citation networks;Narrative views;
 Visual bibliographies;Visual organizers;Visualization;Zz-structures},
 keywords={Bibliographic indexes;Citation networks;Evaluation study;Hierarchical data;
 Narrative views;Real time;Visual organizers;WEB application},
 topics={Software;Management Information Systems;
 Information Systems and Management;Artificial Intelligence},
 year={2019},
 source={Scopus}
}
```

Bibliography

[1] Aminer platform. https://aminer.org/ (accessed on October, 26 2019).

[2] Apps4edu. www.uen.org/apps4edu (accessed on October, 26 2019).

[3] CiteSeerX. http://citeseerx.ist.psu.edu (accessed on October, 26 2019).

[4] Common sense education. www.commonsense.org (accessed on October, 26 2019).

[5] CrossRef platform. https://www.crossref.org/ (accessed on October, 26 2019).

[6] CrossRef platform APIs. https://github.com/CrossRef/rest-api-doc (accessed on October, 26 2019).

[7] Datacite api services documentation. http://doc.aminer.org/en/latest/ (accessed on October, 26 2019).

[8] Digcomp 2.1. the digital competence framework for citizens with eight proficiency levels and examples of use. http://dx.doi.org/10.2760/38842 (accessed on October, 26 2019).

[9] Digital bibliography & library (dblp) project. https://dblp2.uni-trier.de/ (accessed on October, 26 2019).

[10] Ecmascript 2015 language specification. https://www.ecma-international.org/ecma-262/6.0/ (accessed on October, 26 2019).

[11] Edshelf. https://edshelf.com/ (accessed on September, 20 2019).

[12] Educational web apps. educational-web-apps.zeef.com (accessed on October, 26 2019).

[13] Elsevier Scopus APIs. https://dev.elsevier.com/sc_apis.html (accessed on October, 26 2019).

[14] Essediquadro. sd2.itd.cnr.it (accessed on October, 26 2019).

[15] Fetch Live Standard: the CORS protocol. https://fetch.spec.whatwg.org/ (accessed on October, 26 2019).

[16] Free technology for teachers. https://www.freetech4teachers.com/ (accessed on October, 26 2019).

[17] Google Scholar. https://scholar.google.it/ (accessed on October, 26 2019).

[18] Microsoft Academic. https://academic.microsoft.com/ (accessed on October, 26 2019).

[19] OpenAire platform. https://www.openaire.eu (accessed on October, 26 2019).

[20] OpenCitations organization. http://opencitations.net/ (accessed on October, 26 2019).

[21] Orcid platform. https://orcid.org/ (accessed on October, 26 2019).

[22] Orcid platform public APIs. http://members.orcid.org/api/about-public-api (accessed on October, 26 2019).

[23] Scopus platform. https://www.scopus.com/ (accessed on October, 26 2019).

[24] Web Of Science platform. https://webofknowledge.com/ (accessed on October, 26 2019).

[25] Processing. a flexible language to promoted software literacy within the visual arts and visual literacy within technology., 2001. https://processing.org/ (accessed on October, 23 2017).

[26] The film finder application to support the search of films with different types of visualisation., 2004. https://infovis-wiki.net/wiki/Film_Finder (accessed on October, 18 2017).

[27] Jquery, a fast, small, and feature-rich javascript library., 2006. https://jquery.com/ (accessed on July, 16 2019).

[28] A periodic table of visualization methods, 2007. http://www.visual-literacy.org/periodic_table/periodic_table.html (accessed on July, 16 2019).

[29] Flare, an actionscript library for creating visualizations that run in the adobe flash player., 2008. http://flare.prefuse.org/ (accessed on October, 23 2017).

[30] Javascript infovis toolit: interactive data visualizations for the web., 2009. http://philogb.github.io/jit/index.html (accessed on July, 16 2019).

[31] Protovis: a graphical approach to visualization., 2009. http://mbostock.github.io/protovis/ (accessed on July, 16 2019).

[32] D3.js, a javascript library for manipulating documents based on data., 2011. https://d3js.org/ (accessed on July, 16 2019).

[33] Zz-structures - A parallel computer universe, accessed on July, 29 2019.

[34] E. Alexander, S. Bresciani, and M.J. Eppler. Knowledge scaffolding visualizations: A guiding framework. *Knowledge Management and E-Learning*, 7(2):179–198, 2015.

[35] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Proceedings - IEEE Symposium on Information Visualization, INFO VIS*, pages 111–117, 2005.

[36] Lorin W Anderson, David R Krathwohl, P Airasian, K Cruikshank, R Mayer, P Pintrich, James Raths, and M Wittrock. A taxonomy for learning, teaching and assessing: A revision of bloom's taxonomy. *New York. Longman Publishing. Artz, AF, & Armour-Thomas.*, 9(2):137–175, 2001.

[37] P. Bergström and D. C. Atkinson. Augmenting the exploration of digital libraries with web-based visualizations. In *2009 Fourth International Conference on Digital Information Management*, pages 1–7, Nov 2009.

[38] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D^3 data-driven documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–2309, 2011.

[39] S. Bresciani and M.J. Eppler. The pitfalls of visual representations: A review and classification of common errors made while designing and interpreting visualizations. *SAGE Open*, 5(4), 2015.

[40] John Brooke. Sus: a retrospective. *Journal of usability studies*, 8(2):29–40, 2013.

[41] Les Carr. A partial implementation of nelson's zigzag©ideas for web browsers, December, 20 2001.

[42] C. Chen, F. Ibekwe-SanJuan, and J. Hou. The structure and dynamics of co-citation clusters: A multiple-perspective co-citation analysis. *Journal of the American Society for Information Science and Technology*, 61(7):1386–1409, 2010.

[43] L. Chen and H. Zhou. Research and application of dynamic and interactive data visualization based on d3. In *2016 International Conference on Audio, Language and Image Processing (ICALIP)*, pages 150–155, July 2016.

[44] Todd Cherner, Judy Dix, and Corey Lee. Cleaning up that mess: A framework for classifying educational apps. *Contemporary Issues in Technology and Teacher Education*, 14(2):158–193, 2014.

[45] Todd Cherner, Cheng-Yuan Lee, Alex Fegely, and Lauren Santaniello. A detailed rubric for assessing the quality of teacher resource apps. *Journal of Information Technology Education: Innovations in Practice*, 2016.

[46] M. Corbatto and A. Dattolo. A web application for creating and sharing visual bibliographies. *LNCS*, 10959:78–94, 2018.

[47] M. Corbatto and A. Dattolo. Exploring appinventory, a visual catalog of applications for assisting teachers and students. *Multimedia Tools and Applications*, 2019.

[48] Marco Corbatto. Modeling and developing a learning design system based on graphic organizers. In *Adjunct Publication of the 25th Conf. on User Modeling, Adaptation and Personalization*, pages 117–118. ACM, 2017.

[49] Marco Corbatto and Antonina Dattolo. Appinventory: a visual catalogue of web 2.0 and mobile applications for supporting teaching and learning activities. In *Proceedings of the 22nd International Conference Information Visualisation - IV 2018*, pages 530–535, IEEE, July 10-13, Salerno, Italy, July 10-13 2018.

[50] Marco Corbatto and Antonina Dattolo. A web application for creating and sharing visual bibliographies. In *Semantics, Analytics, Visualization – Proceedings of SAVE-SD 2017 and SAVE-SD 2018, Lecture Notes in Computer Science*, volume 10959, pages 78–94, 2018. Springer Nature.

[51] G. Costagliola, A. De Lucia, S. Orefice, and G. Polese. A classification framework to support the design of visual languages. *Journal of Visual Languages and Computing*, 13(6):573–600, 2002.

[52] G. Costagliola, M. De Rosa, V. Fuccella, and S. Perna. Visual languages: A graphical review. *Information Visualization*, 17(4):335–350, 2018.

[53] Gennaro Costagliola and Vittorio Fuccella. Cybis: A novel interface for searching scientific documents. In *Proceedings of the 15th International Conference on Information Visualisation (IV)*, pages 276–281, 2011.

[54] A. Dattolo and F.L. Luccio. A new concept map model for e-learning environments. *Lecture Notes in Business Information Processing*, 18 LNBIP:404–417, 2009.

[55] Antonina Dattolo and Marco Corbatto. Visualbib: Narrative views for customized bibliographies. In *Proceedings of the 22nd International Conference Information Visualisation (IV)*, pages 133–138, IEEE, July 10-13, Salerno, Italy, July 10-13 2018.

[56] Antonina Dattolo and Marco Corbatto. Visualbib: A novel web app for supporting researchers in the creation, visualization and sharing of bibliographies. *Knowledge-Based Systems*, 2019.

[57] Antonina Dattolo and Flaminia Luccio. Formalizing a model to represent and visualize concept spaces in e-learning environments. In *Proceedings of International Conference on Web Information Systems and Technologies - Webist 2008*, pages 339–346, Funchal, Madeira, Portugal, May 4-7 2008.

[58] Antonina Dattolo and Flaminia L. Luccio. A new actor-based structure for distributed systems. In *Proceedings of the 30th Jubilee International Convention: Microelectronics, Electronics and Electronic Technologies, Hypermedia and Grid Systems - MIPRO 2007*, pages 195–201, Opatija, Croatia, May, 21-25 2007.

[59] Antonina Dattolo and Flaminia L. Luccio. A formal description of zz-structures. In *Proceedings of the 1st Workshop on New Forms of Xanalogical Storage and Function, held as part of the ACM Hypertext 2009*, number 508 in CEUR, pages 7–11, Turin, Italy, June 29 2009.

[60] Antonina Dattolo and Flaminia L. Luccio. A state of art survey on zz-structures. In *Proceedings of the 1st Workshop on New Forms of Xanalogical Storage and Function, ACM Hypertext 2009*, CEUR:508, pages 1–6, Turin, Italy, June 29 2009.

[61] Marian Dörk, Nathalie Henry Riche, Gonzalo Ramos, and Susan Dumais. Pivotpaths: Strolling through faceted information spaces. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2709–2718, 2012.

[62] N. Elmqvist and P. Tsigas. Citewiz: A tool for the visualization of scientific citation networks. *Information Visualization*, 6(3):215–232, 2007.

[63] M.J. Eppler and S. Kernbach. Dynagrams: Enhancing design thinking through dynamic diagrams. *Design Studies*, 47:91–117, 2016.

[64] Coral Featherstone and Etienne Van der Poel. Human creativity in the data visualisation pipeline. 2017.

[65] P. Federico, F. Heimerl, S. Koch, and S. Miksch. A survey on visual approaches for analyzing scientific literature and patents. *IEEE Transactions on Visualization and Computer Graphics*, 23(9):2179–2198, Sept 2017.

[66] Tracey Hall and Nicole Strangman. Graphic organizers. *Wakefield, MA: National Center on Accessing the General Curriculum. Retrieved March*, 20:2009, 2002.

[67] Isitgoonair. Mobile learning: apprendimento innovativo attraverso le app ed i dispositivi mobili. `http://mlearning.isitgoonair.net` (accessed on October, 26 2019).

[68] Amaya Jareño, Erla M. Morales-Morgado, and Fernando Martínez. Design and validation of an instrument to evaluate educational apps and creation of a digital repository. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality*, TEEM '16, pages 611–618, New York, NY, USA, 2016. ACM.

[69] José Fernando Rodrigues Jr., Luciana A. M. Zaina, Maria Cristina Ferreira de Oliveira, and Agma J. M. Traina. A survey on information visualization in light of vision and cognitive sciences. *CoRR*, abs/1505.07079, 2015.

[70] Kennedy Kambona, Elisa Gonzalez Boix, and Wolfgang De Meuter. An evaluation of reactive programming and promises for structuring collaborative web applications. In *Proceedings of the 7th Workshop on Dynamic Languages and Applications*, DYLA '13, pages 3:1–3:9. ACM, 2013.

[71] A. Kerren, K. Kucher, Y-F. Li, and F. Schreiber. Biovis explorer: A visual guide for biological data visualization techniques. *PLoS ONE*, 12(11:e0187341), 2017.

[72] Peter Klein, Frank Müller, Harald Reiterer, and Maximilian Eibl. Visual information retrieval with the supertable + scatterplot. In *International Conference on Information Visualisation, London, UK, July 10-12*, pages 70–75, 2002.

[73] K. Kucher and A. Kerren. Text visualization techniques: Taxonomy, visual survey, and community insights. In *2015 IEEE Pacific Visualization Symposium (PacificVis)*, pages 117–121, April 2015.

[74] K. Kucher, C. Paradis, and A. Kerren. The state of the art in sentiment visualization. *Computer Graphics Forum*, 37(1):71–96, 2018.

[75] Bongshin Lee, Mary Czerwinski, George Robertson, and Benjamin B. Bederson. Understanding research trends in conferences using paperlens. In *CHI EA '05 Extended Abstracts on Human Factors in Computing Systems*, pages 1969–1972. ACM, 2005.

[76] Cheng-Yuan Lee and Todd Sloan Cherner. A comprehensive evaluation rubric for assessing instructional apps. *Journal of Information Technology Education*, 14, 2015.

[77] R. Lengler and M.J. Eppler. Towards a periodic table of visualization methods of management. pages 83–88, 2007.

[78] James R Lewis and Jeff Sauro. Can i leave this one out?: the effect of dropping an item from the sus. *Journal of Usability Studies*, 13(1):38–46, 2017.

[79] Deqing Li, Honghui Mei, Yi Shen, Shuang Su, Wenli Zhang, Junting Wang, Ming Zu, and Wei Chen. Echarts: A declarative framework for rapid construction of web-based visualization. *Visual Informatics*, 2(2):136 – 146, 2018.

[80] Gerald L. Lohse, Kevin Biolsi, Neff Walker, and Henry H. Rueter. A classification of visual representations. *Commun. ACM*, 37(12):36–49, December 1994.

[81] Justin Matejka, Tovi Grossman, and George Fitzmaurice. Citeology: Visualizing paper genealogy. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, pages 181–190. ACM, 2012.

[82] Michael J. McGuffin. A graph-theoretic introduction to ted nelson's zzstructures, June 2004.

[83] Michael J. McGuffin and m.c. schraefel. A comparison of hyperstructures: zzstructures, mspaces, and polyarchies. In *Proceedings of the 15th ACM conference on Hypertext and Hypermedia - Hypertext 2004*, pages 153–162. ACM Press, August, 9-13 2004.

[84] Jamie McKenzie. Graphical organizers as thinking technology. *From Now On*, 7(2), 1997.

[85] R.E. Meyer, M.A. Höllerer, D. Jancsary, and T. Van Leeuwen. The visual dimension in organizing, organization, and organization research: Core ideas, current developments, and promising avenues. *Academy of Management Annals*, 7(1):487–553, 2013.

[86] Theodor Holm Nelson. The xanadu® parallel universe. `http://xanadu.com/xUniverse-D6` (accessed on October, 28 2019).

[87] Theodor Holm Nelson. What's on my mind. Invited talk at the *first Wearable Computer Conference*, May, 12-13 1998.

[88] Theodor Holm Nelson. Xanalogical structure, needed now more than ever: Parallel documents, deep links to content, deep versioning, and deep re-use. *ACM Computing Surveys*, 31(4es):1–32, December 1999.

[89] Theodor Holm Nelson. A cosmology for a different computer universe: Data model, mechanisms, virtual machine and visualization infrastructure. *Journal of Digital Information*, 5(1), July 2004.

[90] J.F. Rodrigues Jr., A.J.M. Traina, M.C.F. De Oliveira, and C. Traina Jr. The spatial-perceptual design space: A new comprehension for data visualization. *Information Visualization*, 6(4):261–279, 2007.

[91] Zeqian Shen, Michael Ogawa, Soon Tee Teoh, and Kwan-Liu Ma. Biblioviz: A system for visualizing bibliography information. In *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation - Vol.60*, APVis '06, pages 93–102, Tokyo, Japan, 2006.

[92] Ben Shneiderman. Eyes have it: a task by data type taxonomy for information visualizations. pages 336–343, 1996.

[93] Nees Jan van Eck and Ludo Waltman. Citnetexplorer: A new software tool for analyzing and visualizing citation networks. *Journal of Informetrics*, 8(4):802 – 823, 2014.

[94] Nees Jan Van Eck and Ludo Waltman. Visualizing bibliometric networks. In *Measuring scholarly impact*, pages 285–320. Springer, 2014.

[95] N.J. van Eck and L. Waltman. Software survey: Vosviewer, a computer program for bibliometric mapping. *Scientometrics*, 84(2):523–538, 2010.