SILKNOW

D5.5. Visualization and deployable components

Deliverable number and name:	D5.5 Visualization and deployable components
Due date:	31/12/2019
Work Package:	WP5
Deliverable leader:	UVEG
Authors:	Javier Sevilla (UVEG), Jesús Gimeno (UVEG), Manolo Pérez (UVEG)
Reviewers:	Valeria Seidita (UNIPA), Raphaël Troncy (EURECOM)
Approved by:	Cristina Portalés (UVEG)
Dissemination level:	PU
Version:	V1.0

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769504

Document revision history			
Version	Date	Contributor	Comments
0.1	15/01/2020	Javier Sevilla, Jesús Gimeno, Manolo Pérez	First draft ready to review.
0.2	24/01/2020	Valeria Seidita, Raphaël Troncy	Revised deliverable
0.3	24/01/2020	Javier Sevilla	Applied the changes proposed by reviewers.
0.4	29/01/2020	Daniel Sheerin	English review
1.0	30/01/2020	Javier Sevilla	Produce the final deliverable with the layout

List of acronyms	
UVEG	Universitat de València Estudi General
WP	Workpackage
ICT	Information and Communication Technologies

Table of contents:

Deliverable D5.5 consists of the visualization and 3D printing components, which have been developed under the Task 5.5 as part of WP5, and is of the kind OTHER (software). The visualization components refer to an open and dynamic spatio-temporal data visualization map, while the 3D printing components refer to the 3D virtual models developed in the virtual loom (explained in D5.4), which can be printed. In this document, we explain their design and implementation and give examples of their use. We also explain the relationship with other deliverables and discuss the state of the art.

1. VISUALIZATION COMPONENTS

1.1. INTRODUCTION

SilkViz is a software module that allows the spatio temporal visualization of knowledge graph data. This software uses and expands on Visualization ontology (VISO) [16] work in order to define how the knowledge graph data is going to be visualized. It can be embedded into a web site as a WebGL plugin, or it can be executed as a stand-alone application. In this document, a state of the art in spatio temporal data visualization is presented along with the goals implemented through SilkViz. The intregration with VISO ontology is also detailed as well as the process of visualization stylesheet definition. Finally, the graphical user interface and its API is described.

The first version of SilViz has been developed in the SILKNOW project, thus the 3D models, virtual spaces and geographical limits are highly oriented to the SILKNOW context. In order to expand SilkViz to other projects and solutions, we have introduced the creation of user-defined stylesheets. These stylesheets define how to visualize the data entered with the tool's API . There are two basic scenes, the map scene and the gallery scene. In the map scene, the user can define how to visualize the different classes of data, the region limits, and the possibility of defining time intervals by applying hypercube visualization techniques in order to analyse the data evolution over time in the same region. In the gallery scene, a set of data can be visualized in a user-defined room, allowing the user to walk around and examine the data.

This document is organized as follows: firstly, we describe the relationship with other project deliverables. Next, an overview of the state of the art is given, explaining the novelty of SilkViz, as well as the collaboration with other institutions/experts which has led to improvements. In section 3, the expansion of VISO ontology with the definition of visualization stylesheets is described. Section 4 is devoted to the design and implementation of SilkViz, which includes the graphic user interface and its API. In section 5, some examples and results after processing with SilkViz are shown. The last sections are the conclusions and references.

1.2. RELATIONSHIP WITH OTHER DELIVERABLES

The content of D3.2 “Design of SILKNOW ontology and the ontology" is very important for the development of SilkViz, as the ontology defines the data structure of the knowledge graph which SilkViz is in charge of visualizing .

In D6.3 “Ontology web server” the API of the Ontology web server is implemented and described. This API will be used to get the data which is going to be visualized in SilkViz. The visualization tool is independent from any visualization platform tool, but in the SILKNOW project the data must be gathered from this tool and its API must be used in some way.

1.3. STATE OF THE ART

The visualization of information helps the process of obtaining knowledge provides a better understanding of the information through the graphical representation of massive data sets [4]. In the case of spatio-temporal models, the represented data fluctuate in space and time. They are data sets that are difficult to visualize, especially given the other dimensions, in addition to space and time [5] , which are present. In order to achieve this objective, the data have been structured in different levels over the last two decades. A series of operations are defined for each level to be performed. These classifications define what can be queried in the system and what can be obtained from each query. These studies have been carried out in conjunction with the evolution of the tools and user interface elements which allow data to be consulted as well as viewed. Next, an analysis of some of the most widespread classification techniques is given, as well as the different types of developments.

1.3.1. Classification of Spatio-Temporal Data

Peuquet talks about the existence of three main parts in spatio-temporal data: where, when and what [18]. These parts describe a location, a time and the data to be represented. To the same type of task-oriented model, more extensions were added; for example, Andrienko [19] classified the tasks to analyze the information consulting points, regions and trajectories. With the evolution of technology, the data has changed, the size of the information is much larger and there are many more attributes to consider. It is important to consider objects and their attributes. Nowadays, with massive amounts of available data, more attributes can be considered, helping with the identification of patterns.

Level	Tasks contained in each task level
1	where + when + who → what where + when + what → who where + who + what → when who + what + when → where
2	where + when → what + who what + when → who + where where + what → who + when when + who → what + where where + who → when + what who + what → where + when
3	where → what + when + who what → who + when + where when → where + who + what who → where + what + when
4	NULL → what + where + when + who

Figure 1. List with the tasks contained in each task level. Source: Guo [12].

For this reason, new ways of classifying data arise, such as that proposed by Guo [12], which adds an additional part to the three proposed by Peuquet, the "who", which refers to objects, or series of objects. In addition, Guo associates the "what" part with the attributes of these objects. The four parts can be combined around four levels, performing queries with a greater to a fewer number of variables, so that the fewer variables used, the more information will be shown, reducing the certainty of the analysis. A table with the different tasks for each level can be seen in Figure 1. These new ways of classifying data generate new results that allow a better analysis of the data and the discovery of relationships, or patterns, in an increasingly simple and rapid manner. However, since reality is very complex, new visualization demands appear as the system improves.

A new requirement appears when objects are both objects and attributes. This happens when an object is related to other objects. In this case, the related objects are new objects and attributes of the source object. An example would be manufacturers and manufactured objects which evolve over space and time. It is very interesting to see where these objects have been manufactured, but also where they have been used, and what they are related to. If we want to see the development of this evolution in space and / or time, the problem is even greater and requires further improvement of techniques.

1.3.2. Visualization tools

From the beginning of this research, many software tools have been developed to visualize information [1] [2] [4]. Over the last two decades, due to technological advances, the use of three-dimensional representations with a high level of interactivity has increased.

Some developments show information (what + where + when) over a bi-dimensional or three-dimensional region. These tools use various techniques such as colour maps, colour temperature patterns and clustering techniques [22][23]. Colour, cluster size, the number of dimensions (2D/3D), elevation, and even the use of additional labels, are the visual elements that are used to represent information in the graph (Figure 2).

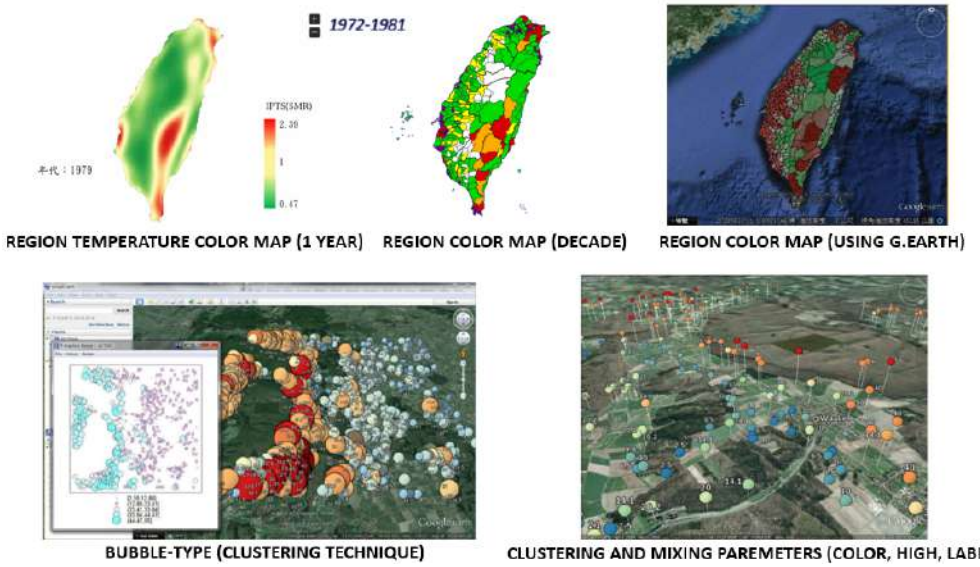


Figure 2. At the top, screen shots of a web application that uses coloured regions and colour temperature maps to visualize the incidence of cancer mortality in Taiwan. Source: [20]. Below, screenshots from tools that use clustering techniques to represent the information. Source [21].

In addition to visualizing information and displaying common form controls in the user interface, innovative controls have also been created. These controls are used in order to define and filter the time in which the information to be displayed is represented. One of the most used interactive devices is the timeline [24] [25], which contains information concerning the different time levels. It starts from a broad level, usually in the upper part, going down interactively until selecting a concrete period in the lower zone. It is often used in cultural heritage visualization tools [26]. See Figure 3 for examples of timelines, specificallya Time Wheel control [19] which allows the selection of specific months, as well as days and a time interval in a simple way.

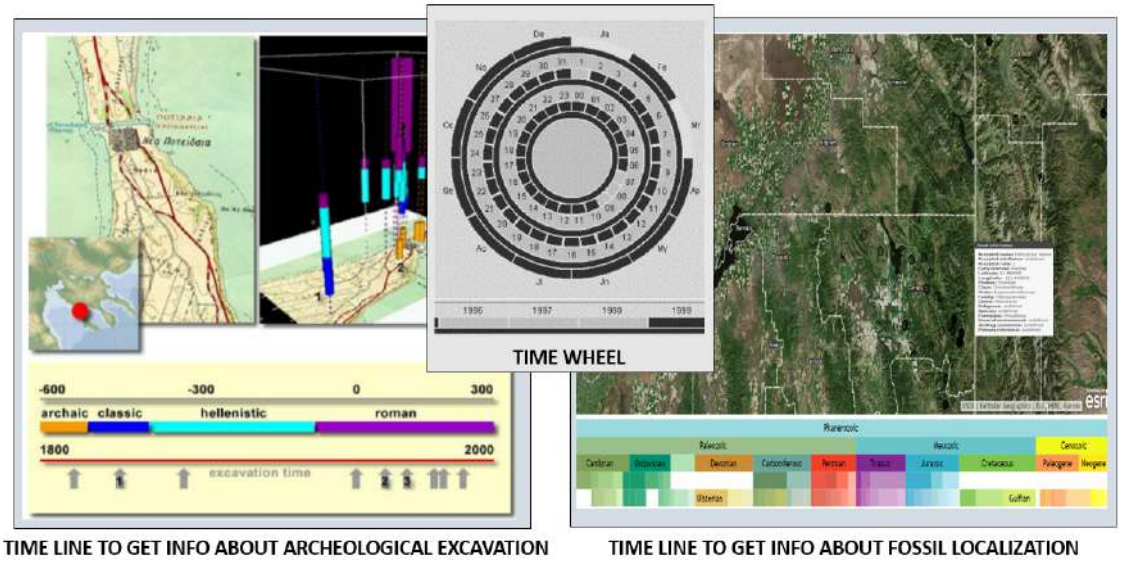


Figure 3. Left and right show screenshots of spatio-temporal data visualization applications in a cultural heritage application. Source: [24] [26]. Screenshot of a Time Wheel control in the upper central part. Source: [19].

Analysing how information evolves over time can be the main goal in determinate information systems. In this case, it is important to see the date at different time intervals. To solve this problem, the scientific community has also developed several solutions. Some are based on three-dimensional spaces, like space-time cubes [5], or spirals [27], which display a 3D spiral on the object, where each lap represents a period of time. A spiral shows a lot of information, but it is very difficult to analyze. Others are based on two-dimensional spaces, like flow maps [28], used to represent trajectories or the movement of objects over time. Finally, animations are also used where each frame corresponds to a specific period. Some applications that use these techniques can be seen in Figure 4.

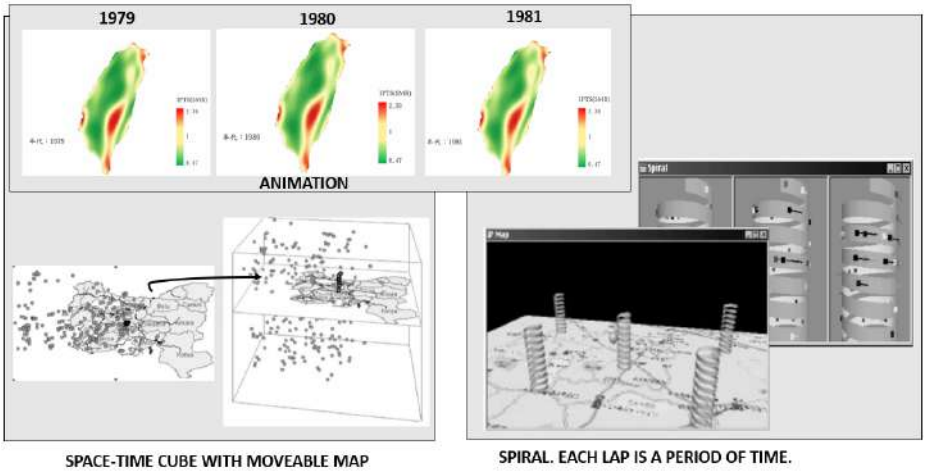


Figure 4. The upper part of the image shows the different frames, one per year, of the incidence of terminal cancer in Taiwan, resulting in an animation. Source: Adapted from [20]. In the lower left zone, a space-time cube is displayed, with a horizontal plane (map) which can be moved. Source: [19] To the right, two screenshots of an application that uses 3D spirals, where each lap is a period of time and has additional information related to the appropriate year. Source: [27].

The advance in the use of these techniques has been important, especially in the graphic quality and the possibilities of interaction, because many of these ideas were developed several decades ago. One of the main problems in this area is that developments are often linked to a particular application. For this reason it is difficult to reuse the tool, and frequently the only reused components are the web services used to show maps, or satellite images, such as Google Maps and others [29].

1.3.3. Application of Ontologies in Visualization Systems

Typically, Big Data tools process information without the need for high semantic content. For instance, there are image-processing tools where the content of images is analysed using neural networks, looking for patterns in the image, and taking as a reference a large sample of correctly classified images [30]. Probably, if semantic information was available, the process would be faster, because this information could contribute to avoid higher computational cost processes.

Ontologies are used to represent complex models of information and to infer knowledge from them, because a high semantic content can be defined in the representation. In the 2014 Ontology Summit [6], it was stated that although ontologies have played a key role in Semantic Web technologies, their application in new lines, such as Big Data, is lower than expected. It seems logical that both lines of research should have had a greater relationship, but, in fact, this has not happened. Seemingly, the measures proposed in the 2014 Ontology Summit to increase its application in these types of projects are beginning to yield results over recent years [8]. New projects that combine the results of these two lines are in operation [9] [7] [10], and the tools that allow access to the data of an ontology are also evolving [31] [32]. If work along these two lines follows the expected path, the information to be visualized, or at least an important part of it, would be in an ontology. Therefore, the visualization of spatial and temporal data in these types of projects could take advantage of the results obtained in the graphical visualization of ontologies. However, the main activity of this field consists of the visualization of the structure and content of the ontology by means of interactive

representations of graphs, treemaps and similar techniques [13] [33]. In these works, the main objective is usually not the analysis of content in order to discover patterns or to deduce complex relations. For this reason, these investigations may provide components that are useful in the study of spatio-temporal data, but they have not been developed as much as the techniques used in this line of investigation.

In the last decade, new research proposes the use of an ontology to specify how its content should be visualized, as well as to define the user interaction [14] [15] [16]. One of the most generic and reusable results is Visualization Ontology (VISO) [16]. This ontology allows the definition of its concepts, considering how to visualize them and how to interact with them. This ontology was developed as the basis of the RDFS/OWL Visualization Language (RVL) [17].

The problem with VISO ontology is that it involves close consideration of visualization aspects in a two-dimensional environment, but hardly delves into 3D visualization components. It would be necessary to create new classes and extend others to consider three-dimensional representation, which is fundamental in many spatio-temporal data visualization techniques.

1.4. DESIGN

1.4.1. Considerations

Once the state of the art has been revised, the following issues and conclusions arise regarding the visualization and analysis of spatio-temporal data:

- More work needs to be done on the visualization of the relationships between different objects- objects that are at the same time objects and attributes.
- It is important to create more open and multi-platform tools that can be used to represent and analyse spatio-temporal data. These tools must be capable of being embedded into other projects, especially web applications.
- The use of ontologies to represent the information to be visualized should be encouraged. In addition to values, data can contain semantic information, allowing new possibilities of interaction and visualization.
- Generic tools have been created in order to define how the information is to be displayed. VISO ontology, one of the most advanced, needs to be extended to more deeply contemplate the needs of graphics in three-dimensional environments.

In order to satisfy these needs, we propose carrying out the following tasks in the implementation of the SilViz tool:

- To eliminate limitations in the three-dimensional environment of VISO ontology and the RVL language, as well as including activities and interactions in order to implement techniques for the representation of spatial and temporal data.
- To define a way to visualize relationships between objects, which allows the visualization of the evolution of these relationships at different moments of time, simultaneously.
- If the data are represented by an ontology, use VISO ontology and the RVL language to define how to visualize the information.
- To develop an open source software tool in a multi-platform development system that allows the visualization of the results of any project, especially on web platforms.

The selected multi-platform development system is Unity. Unity is the most popular game engine. It has been around since 2005, and it has developed a massive following and an amazing library of resources. Unity allows the generation of code in many operative systems: Windows, Mac OS, and Linux, and in addition in a website through a WebGL Plugin. The Unity Asset Online Maps[1], from Infinity Code, was purchased in order to manage the online maps tile texture sources downloading process. The source code of this asset can not be freely distributed, and only SILKViz developers need to aquire this asset. End users, who are only interested in visualizing their data, do not need the source code.

Next, we describe how the visualization of relationships is performed, and also the application of the VISO ontology results.

1.4.2. Visualization of object relationships

In the visualization of spatio-temporal data, the representation of different objects related to each other is one of the most complex activities. It is not possible to take advantage of the work carried out in the visualization of graphs, or trees, at least not in a general way, because in this model, location and time are fixed values, and therefore it is not possible to move them to improve their visualization. When there are many objects, the classic solution consists of developing clusters of objects, visualizing the relationship of an object with a cluster of objects, or relationships between different groups.

Flow maps [28] usually include trajectories. These visualization systems have worked with this type of techniques in the relationships between objects. This approach reduces the simultaneous number of connections, and facilitates visualization and comprehension. However, in spatio-temporal data, groupings usually have a spatial base and, despite using these techniques, the number of connections to be visualized usually prevents the correct visualization. A typical solution is interaction: selecting an object, or group of objects, and displaying only the relationships of the selected object. An application that solves this problem by visualizing the object relationships interactively is shown in Figure 5.

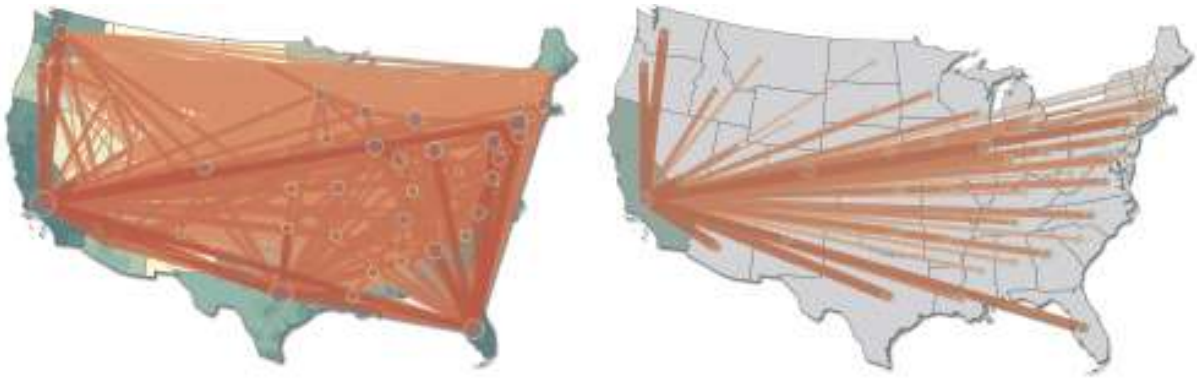


Figure 5. Visualizing all the relationships generates images that are very difficult to analyze. Visualizing the relationships of a single object is more readable. Source: [28].

Using interaction has important limitations: a previous lack of knowledge concerning whether an object is related or not, as well as the number of its relationships. However, investigating those objects with a large number of relationships, different from the average, is generally

[1] Unity Asset Store Home Page: https://assetstore.unity.com

more interesting. This problem becomes more difficult to handle when we want to study several elements simultaneously, as was detailed in the state- of- the- art section.

The use of marks around the object, or clusters of objects, is proposed which identify the level of relationships compared with the average. The mark would consist of a ring that would enclose the object, divided into as many sectors as objects it is related to. Longer or shorter lines would be drawn on this sector, depending on the number of relationships. On the other hand, the length of the sector is associated with the time interval. Thus, the beginning of the sector will be the beginning of the period, and the end of the sector, the end of the interval. Each sector is a histogram which is encoded into a circle. A diagram showing the ring marker and the final appearance on a map can be seen in Figure 6. This marker will be implemented in the tool and mapped using VISO.

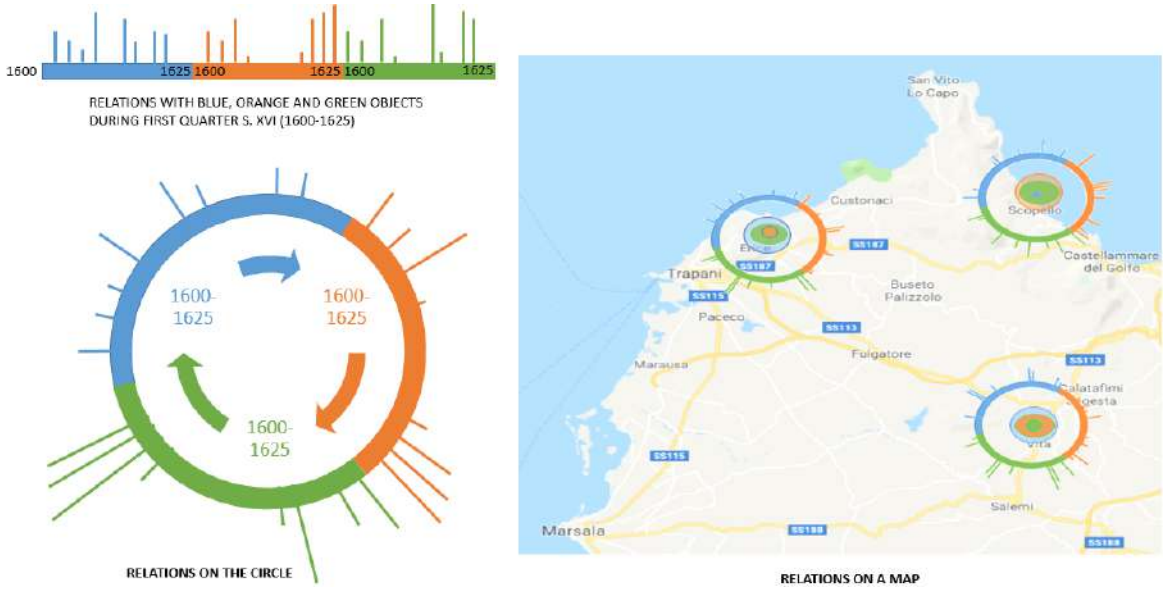


Figure 6. Ring mark to show information about the number of relationships that each object/cluster has with other objects/clusters during the selected time. To see the relationships, you would have to select one of the objects.

1.4.3. VISO ontology application

Visualization Ontology (VISO) formalizes knowledge from the domain of visualization in order to make it usable by machines, and allow for an exchange between tools and users. Machine-readability and interoperability is achieved using well-established Semantic Web standards, such as RDF(S) and OWL [2]. VISO is modularized into seven parts (Figure 7). The most important modules are:

- The GRAPHIC Module, which formalizes graphics.
- The DATA Module, which formalizes data variables and structures.
- The ACTIVITY Module. This module is concerned with the human aspects of visualization, for instance actions and operations.
- The SYSTEM Module, the USER Module and the DOMAIN Module. These modules allow the description of the visualization context and domain-specific facts.
- The FACTS Module, which formalizes constraints and rankings.

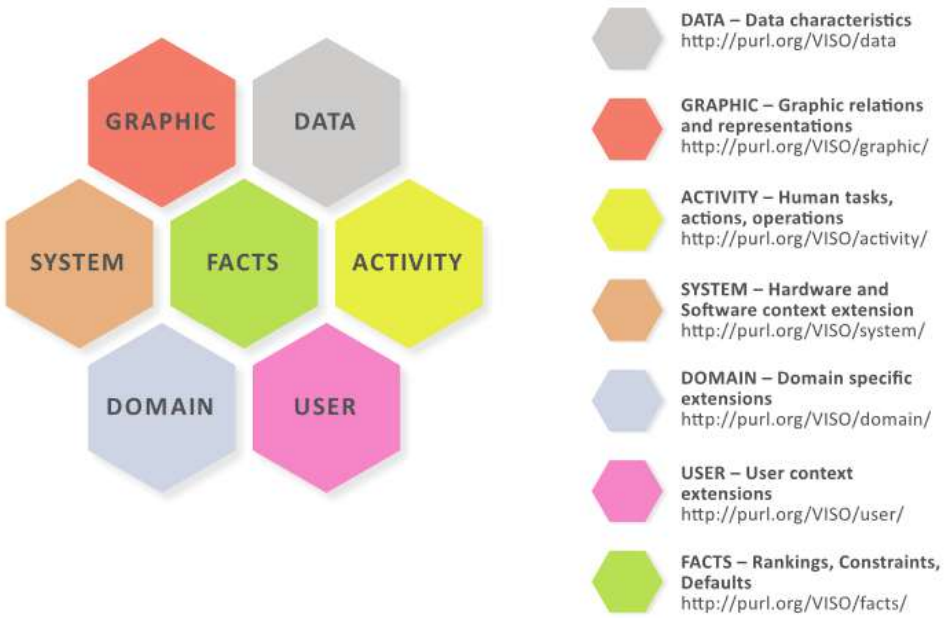


Figure 7. VISO ontology is made up of seven modules. Each module is related to a different visualization field.

VISO ontology is a fantastic way to formalize any visualization system, but it was created to represent how to visualize the ontology data in a device independent way. This is a great idea, but it has problems. One problem is related to this device independence, which avoids there being a concrete specification about how to visualize the data. For instance, very important aspects like position, height, width, scale or resolution are not included in the formalization. The authors of this ontology proposed the definition of specific stylesheets to define those and other concrete details. Each specific visualization will use its stylesheet in order to adapt to the device. In addition, VISO ontology is highly oriented to a 2D visualization, hence it is necessary to extend the current classes in order to formalize 3D graphic visualization.

Table 1 shows the main extensions carried out in VISO ontology to manage 3D graphic visualization.

New class	Classes extended	Properties used
Graphic Representation 3D	Graphic Representation	
Graphic Object 3D	Graphic Object	File Primitive Colour
Interactive Graphic Representation 3D	Interactive Graphic Representation	Tilt_allowed
Map3D	Map	
DynamicMap3D	Map3D, Interactive Graphic Representation 3D	Clustering_allowed, Data_allowed
Cluster	Graphic Object	Clustering (Spatial, Shape) Domain Ring
Cluster3D	Cluster, Graphic Object 3D	

TimeMap	DynamicMap	Levels
TimeMap3D	TimeMap, DynamicMap3D	
Space3D	Space	Borders
Gallery	Space	Elements Disposed (Free, Rows)
Gallery3D	Gallery, Space3D	
Point	DataStructure	X Y
SpatialData	Point	
TimeData	Point	
RelationRing	N-AryGraphicO2ORelation	

Table 1. Main class extensions performed on VISO ontology to formalize 3D graphics.

In order to visualize data, mapping from data objects to graphic objects is required. The VISO ontology creators propose the use of the RVL/OWL visualization language. RVL defines a set of mapping types that can be instantiated to describe visual encodings from relationships, classes and individual values in the source data to graphic concepts.

SILKViz visualization will allow differents aspects to configure it (3D models, navigation limitations, zoom levels, gallery exhibitors, etc.), but it does not make sense to allow the definition by adding configurable elements. The majority of SilkViz users prefer the modification of a reduced set of items, instead of using a very complex mapping language. Due to this, every visualization is related to the Scene concept. The scene configuration items are mapped to VISO classes in order to take advantage of VISO, but the way to navigate and interact is mainly intrinsic to each scene. In this document, in the API section, the Scene configuration structure will be described. This idea is depicted in Figure 8.

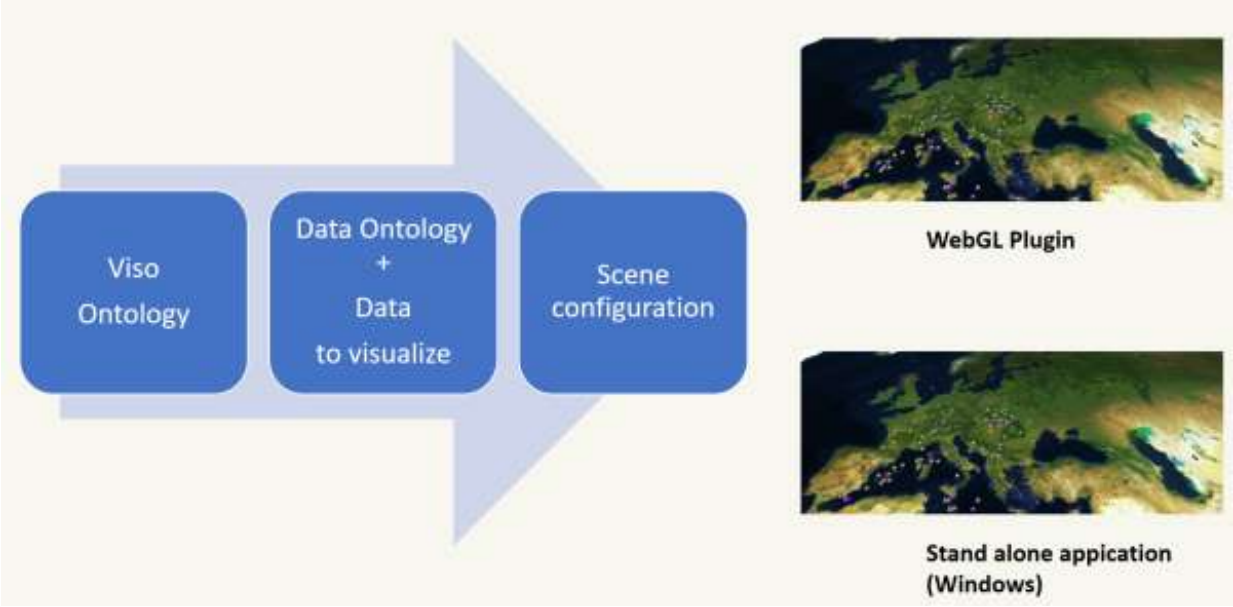


Figure 8. SILKViz input data is the same for stand-alone applications and for the WebGL plugin.

1.5. IMPLEMENTATION

The SILKViz tool is capable of:

- Representing data on a region. This region could be a city neighbourhood, a country, or the whole world, hence it must be capable of getting maps with differents resolutions about the whole world.
- Managing a variable set of data from reduced sets to large amounts of data. Due to this, sometimes the use of spatial clusters to visualize data in an understandable way would be necessary.
- The data aspect must be user configurable (data, spatial clusters, etc.)
- Allowing the visualization of different classes of data.
- Displaying the relationships (if one exists) of the represented data on the map. Showing all the relationships is normally not a good idea, because this would make it very difficult to easily understand the content of the map.
- Representing the fluctuation of data over a time period. Allowing the user to define a time partition and visualize a map with the corresponding data for each interval (hypercube).

1.5.1. Map texture source

In order to be capable of representing a map of every city in the world, it is necessary to have access to a map respository. SILKViz allows the user to access different map repositories: by default, the Opentreemap respository is used. This tile source is freely available and,in addition, if the tiles are used in an open data project, like the SILKNOW project, the tiles can be saved for free on a private server.

The user can access other repositories by changing the value of the DynamicMap.type, which is shown in the API section. The user can have free access to Opentreemap data, directly from Opentreemap services, their own server, Mapsurfer or OpenTopoMap. In addition, repositories, which are not free, are accessible, such as Google, ArcGis, or Mapquest, but in this case the user must pay a fee to the data provider, normally based on the number of requests performed.

In Figure 9 the same map and data with different tile providers is shown.

Figure 9. In this figure the map visualization using different tile providers is shown. (From left to right, Mapquest, Opentopomap and Opentreemap).

1.5.2. Spatial clustering

In order to manage large amounts of data, the automatic creation of spatial clusters on the map is necessary. A spatial cluster represents a set of data that is located in the same area. If in the same area there are a set of points, the user sees the cluster marker instead of all the point markers. In a dynamic navigational map, the screen area dimension changes as the user zooms in or out. In this way, each zoom level has different areas, thus also different spatial clusters. SILKViz manages this situation by using quadtrees (See Figure 10). The user may define the dimension of the area associated with a spatial cluster by the specification of:

- The number of zoom levels of the map. On one hand, there is the zoom level of the map texture, and on the other hand the levels managed from the clustering process. This last value is specified using the property DynamicMap.zoom_levels, described in the API section.
- The levels where the clustering process is activated. If the user is navigating inside these limits, the clusters markers are displayed on the map instead of the point markers. These values can be set using the properties: DynamicMap.ClusterFrom and DynamicMap.ClusterTo. If the clusters are displayed in all the levels, then the point markers are also displayed in all the levels.
- The numbers of quads (areas) in the top zoom level. Every quad in the top level is divided into 4 quads in the next level area, and so on (see figure 10). This value is set using the property DynamicMap.NumClusters.

The size of the cluster marker representation is greater based on the number of associated elements. This feature is independent of the marker representation, a GraphicObject3D (3D model or primitive). In Figure 11, a map with two kinds of cluster markes and the point markers is shown. This representation is confusing, and normally the clusters and the point markers are not represented simultaneously.

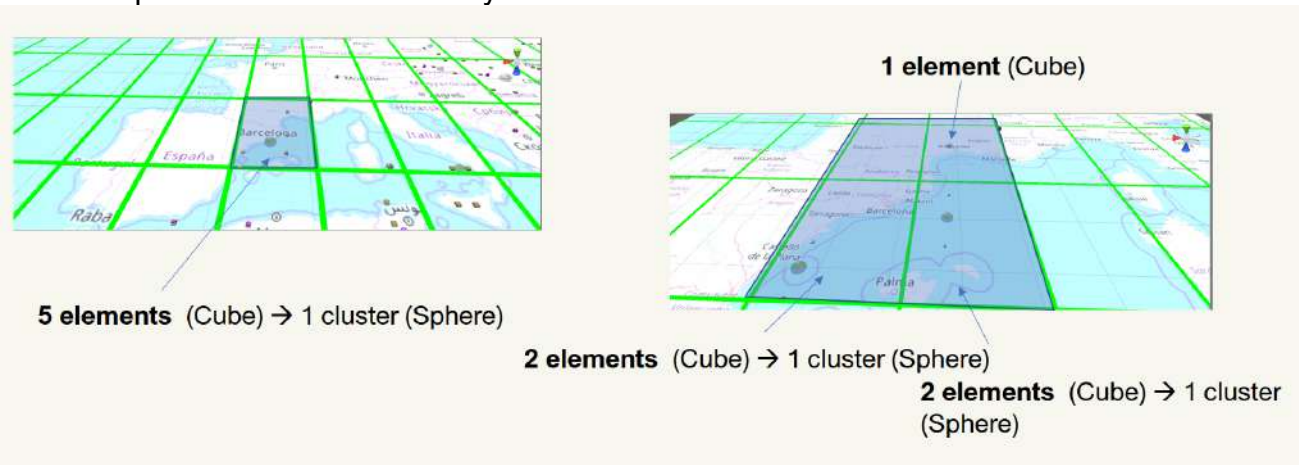


Figure 10. The area associated with a cluster in a zoom level is divided into 4 areas in the next zoom level. On the left there is a quad with 4 point markers (cube) and 1 cluster marker. On the right, the same area is divided into 4 areas, and there are quads without clusters and point markers.

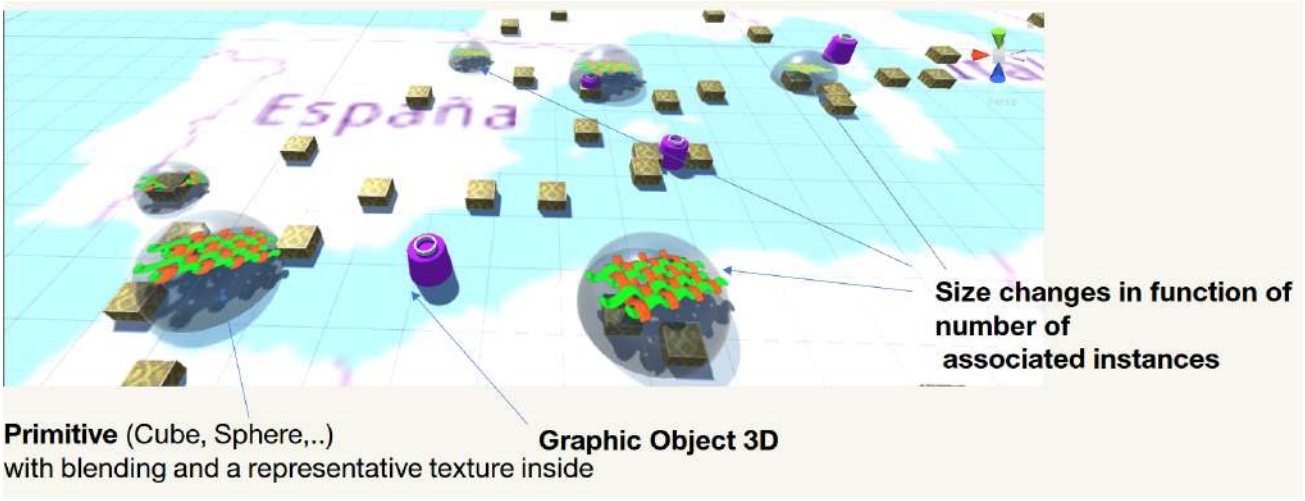


Figure 11. The map markers are 3D models or Primitive (Sphere, Cube, etc.) The cluster markers become larger as the number of represented elements increases.

1.5.3. Display objects relationships

Visualizing the relationships between the different objects displayed on the map is very important. By studying these relationships, researchers can discover new information, which is difficult and hard to see just by analyzing reports with numeric data. As discussed previously, seeing all the relationships simultaneously on the map is very problematic normally there is a large amount of graphical data, and it is challenging to separate one relationship from another. Due to this, SILKViz allows the user to see the relationships of one specific object, by clicking on it, if the adequate property in the configuration data is established (DynamicMap.Relationships), such as its colour, and also the presence of Ring graphical structures. With Rings, the user knows if there are relationships with other objects, and also the number of relationships without having to click on the object. Thus, by looking at the map the user knows where there are more or fewer relationships than the majority. If he/she wants to know the related objects, then it is only necessary to click on the marker and the relationships are displayed. In Figure 12, some clusters with a Ring are shown, and the relationships of each object also are depicted.

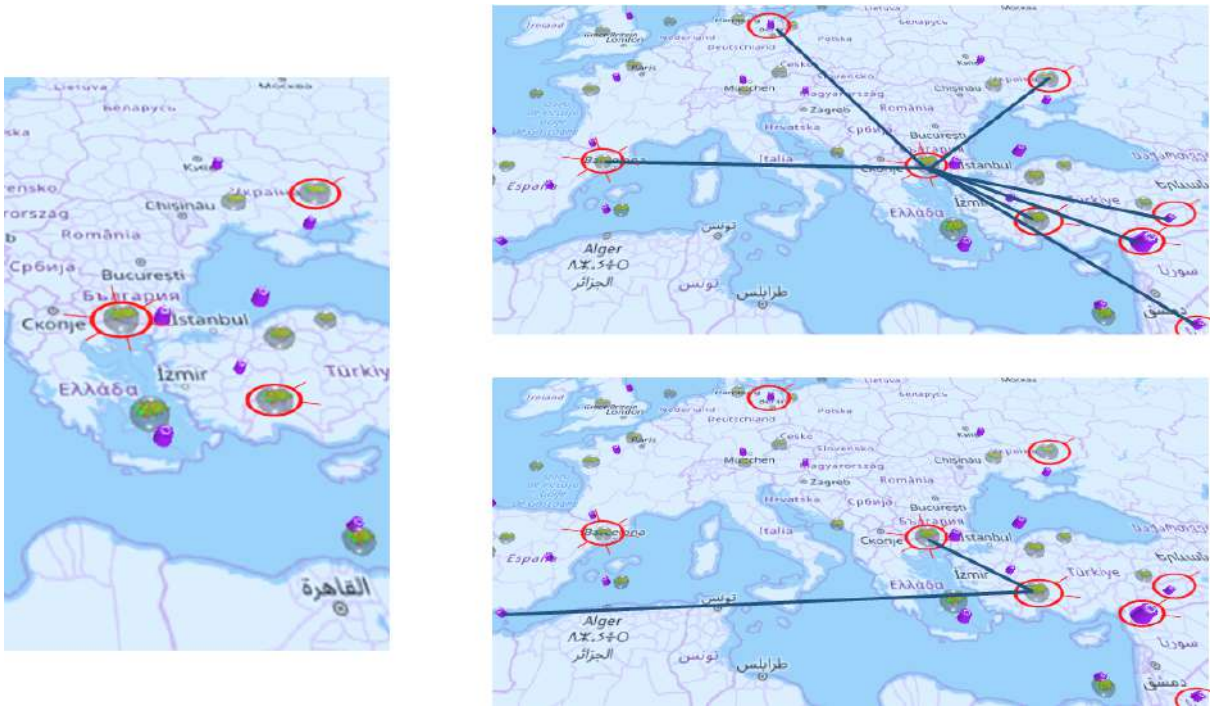


Figure 12. Ring structure and relationships. On the left are clusters with a Ring, which shows how many relationships the cluster has. On the right the relationships of every cluster are shown.

1.5.4. Visualization of additional scenes and data

In order to see the data associated with a point marker the user only needs to click on it. By clicking on the marker, a tooltip is displayed on the screen with the basic data: name, temporal and spatial data. Additional data related to the object can be accessed through a link. In Figure 13 an example of this tooltip is shown.

Figure 13. A tooltip with basic data is presented by clicking on the point marker.

With SILKViz, in addition to the visualization of data on a map, the user can see the content of a cluster in another virtual space, called the Gallery. This is like a museum, where the user can move around looking at the data content. SILKViz has a default gallery scene, but users who work with SILKViz can provide 3D models for the walls, floor, roof and exhibitors: they also can specify if the data is freely exhibited or follows a specific order. In Figure 14, a screenshot with an example of the gallery scene is shown.

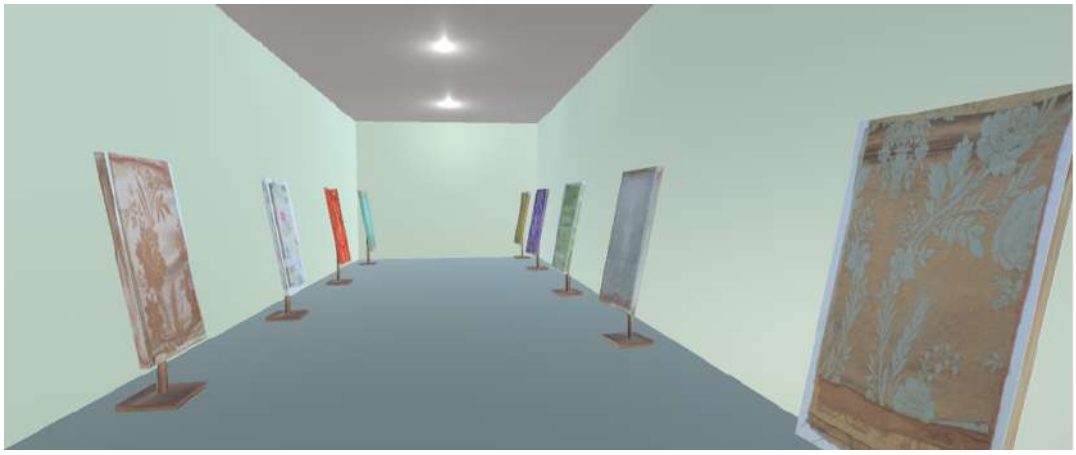


Figure 14. Screenshot of the scene gallery.

1.5.5. Displaying time fluctuation data

Up to now, we have described how SILKViz allows the visualization of spatial data. Clustering, relationships, rings and the use of three-dimensional graphics in real-time are some of the strategies used by this software tool. In this section, we will explain how spatial possibilities are managed, and how is visualized the data evolution over time.

All the data visualized on the map are in a temporal interval. SILKViz offers the possibility of splitting this interval into 2 or 3 fractions, in order to see the data distribution per time interval. This action is performed by displaying a map per time interval. Each map has a different three-dimensional elevation, thus all the maps can be shown on the screen. In addition, the user can modify the perspective of one of the views in order to obtain a better visualization experience in that view. In Figure 15, three maps are shown at the same time, and it also shows how the perspective of the central map can be changed.

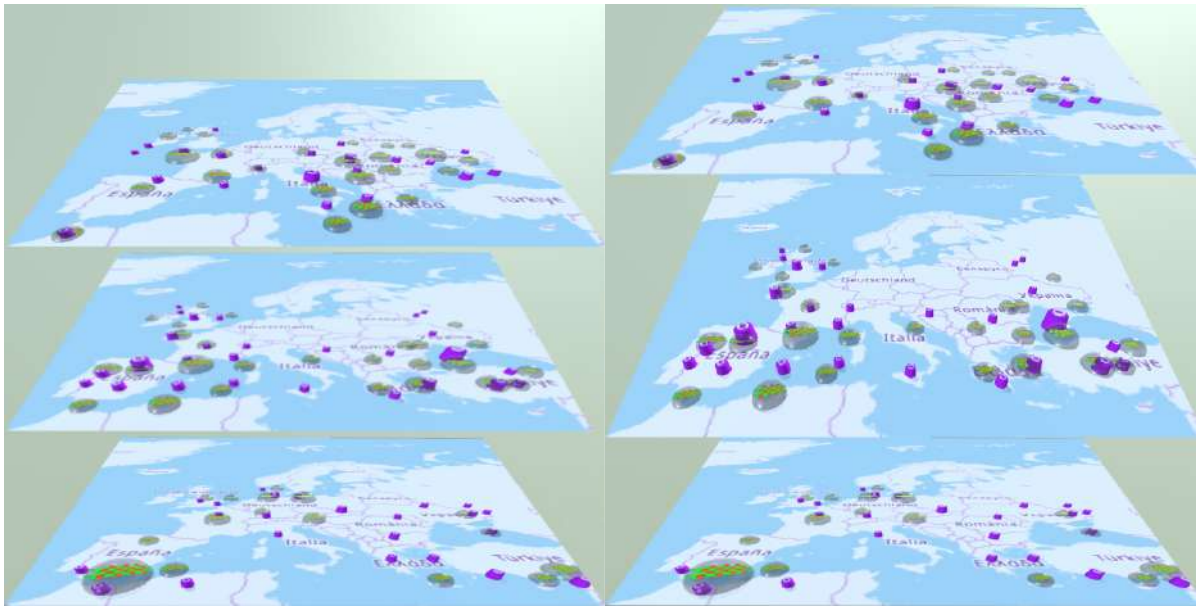


Figure 15. On the left three maps are shown at same time. The temporal interval is the XVI century. Each map is associated with a third of that century (1500-1533, 1533-1566, 1566-1600). On the right you can see how the perspective of the central map changes allowing for a better visualization.

To represent the data relationships, not only in the same temporal interval but also at different time periods (other maps), the user can click on a cluster, and lines are displayed on this map, but on the other maps the objects are highlighted. The initial idea was to display lines between different maps, but the visualization was very difficult to understand.

1.5.6. User interface

The user interface is very important in the visualization of three-dimensional representations. In the SILKViz tool there are two ways to interact: by using the mouse, or by using the keyboard. Keyboard interaction is very important because an external device could emulate it, allowing the accessibility of the tool.

The mouse controls are:

- By clicking the left button and dragging the mouse, the user can move the visualization in the same direction.
- By moving the mouse wheel, the user can increase or decrease the zoom level.
- By clicking on the cluster, its relationships are represented.
- By clicking on the point marker, its data are displayed.
- By clicking the right button, the gallery scene is shown.
- By clicking on the time button, a menu appears in order to choose the number of superposed maps.

The keyboard controls are:

- By pressing the cursor keys, the user can move the visualization in the cursor direction.
- By pressing the “+” and “-“ keys, the user can increase or decrease the zoom level.
- By clicking on the “Enter” key, the relationships of the nearest cluster to the center of the screen are displayed.
- By clicking on the “Enter” key, the data of the nearest point marker are shown.
- By clicking on the “.” key, the gallery scene of the nearest cluster to the center of the screen is shown.

1.6. SETUP

The SILKViz tool could be used as a stand-alone application by executing a Windows operative system executable file. In this case, the user only needs to modify the configuration file and execute it.

The other way to use SILKViz is by embedding a WEBGL plugin into an HTML web page. In this option, the web developer must include the folders “Build” and “TemplateData” in the web server. Once these folders are saved, he/she needs to follow the instructions below.

In the head section of the HTML Web Page:

```
<link rel="shortcut icon" href="TemplateData/favicon.ico">
<link rel="stylesheet" href="TemplateData/style.css">
<script src="TemplateData/UnityProgress.js"></script>
<script src="Build/UnityLoader.js"></script>
<script>
  var gameInstance = UnityLoader.instantiate("gameContainer", "Build/webgl.json",  {onProgress: UnityProgress});
```

```
  </script>
```

Note that the user preferences web location for the “Build” and “TemplateData” folders (bold) must be typed.

In the place where SILKViz is going to be visualized:

```
<div class="webgl-content">
    <div id="gameContainer" style="width: 960px; height: 600px"></div>
    <div class="footer">
      <div class="webgl-logo"></div>
      <div class="fullscreen" onclick="gameInstance.SetFullscreen(1)"></div>
      <div class="title">silkmap0</div>
    </div>
  </div>
```

1.7. CONFIGURATION AND API

There are two ways to execute the SILKViz tool: by running a stand-alone application and by embedding aWebGL plugin into a HTML web page.

In both cases, SILKViz needs the same configuration file: SILKVIZ.JSON

This file must contain the data ontology URL, the visualization ontology URL and the UR; with the stylesheet or scene configuration file. Optionally, the configuration file may contain a visualization data URL, but this is not mandatory. If there are no data to represent, SILKViz will display a whole world map.

This is the default content of SILKVIZ.JSON

```
{
 “DataOntology”: “http://data.silknow.org”,
 “VISOOntology”:”http://purl.org/viso/”,
 “StyleSheet”:”style.json”
}
```

DataOntology refers to the data ontology.
VISOOntology refers to the visualization ontology.
StyleSheet refers to the stylesheet scene configuration.

If there is a data file, the user can enter this using the key “**DataFile**”.

```
{
“SceneList”:{
“Scene”:{
            “name”:”<name>”,
            “based”: “<URI_DynamicMap>”|”<URI_Gallery>”|”<URI_TimeMap>”,
            “mapData”:{ // If based on DynamicMap or TImeMap
                        “dataSource”:”<data_Source_key”>”,
                        “zoomLevels”:”<zoom_Levels_on_map>”,
                        “views”:”2D|3D|All”,
                        “clusters”:{
                                    “fromLevel”:”<level_from_the_clusters_will_be_shown>”,
                                    “toLevel”:”<”level_to_the_clusters_will_be_shown”>,
                                    “fromDataLevel”:<level_from_the_point_markers_will_be_shown>”,
```

```
                        “numQuads”:”<num_of_quads_for_clustering_on_the_top_level>”},
                “timeIntervals”:”2|3|4”
        }
        “galleryData”:{ //if based on Gallery
                borders:{
                        “wall0”:{ ( The same for “wall1”,”wall2”,”wall3”, “roof”,”ground”)
                                “Model”:”<URL_FILE>”,  // If no content is represented a primitive
                                “Color”:”<color>” // If model is empty.
                        },
                exhibitor:{
                        “model”:”<URL_FILE”>, //If no content is represented a primitive.
                        “color”:”<color”>, // If model is empty,
                        “disposed”:”<rows|free>”
                }
        }
“ClassesList”:{
        “Class”:    {
                “URI”:”<URI>”,
                “SceneConfiguration”:{
                        “SceneName”:”<name_of_the_scene>”,
                        “PointRepresentation”:”<URL_MODEL>|Sphere|Cube|Cylinder”,
                        “PointColor”:”<color>”,
                        “ClusterRepresentation”:”<URL_MODEL>|Sphere|Cube|Cylinder”,
                        “Spatial_Data_Properties”:{
                                “longitud”:”<longitud_name_on_DataFile>”,
                                “latitud”:”<latitude_name_on_DataFile”
                        }
                        “Time_Data_Properties”:{
                                “from”:”<from_name_on_DataFile>”,
                                “to”:”<to_name_on_DataFile>”
                        }
                        “nameProperty”:”<name_on_DataFile>”,
                        “relatedToProperty”:”<relation_name_on_DataFile>”,
                        “relation”:{
                                “based”:”< URI_Direct_Linking>”,
                                “color”:”<color>”
                        }
                }
}

}
```

The DataFile format is as follows:

```
{“PointList”{
        “URI”:”URI”,
        “class”:”<URI_CLASS>”,
        “name”:”<name>”,
        “from”:”<from>”,
        “to”:”<to>”,
        “longitud”:”<longitud>”,
        “latitud”:”<latitud>”,
        “image”:”<URL_IMAGE>”,
        “relationWith”: {       “URI”:”<URI>”}
        }
}
```

2. 3D PRINTING COMPONENTS

2.1. INTRODUCTION

The 3D printing components allow to transform the 3D virtual models into tangible forms. These components are integrated as part of the Virtual Loom as a functionality. The 3D printing components allow the user to adapt the 3D models shown on the application for the printing in a standard 3D printer, after sporting a 3D solution in STL format (Figure 16).

Figure 16. 3D printing components integrated in the Virtual Loom. The “Generate STL File” button is on the bottom right.

2.2. RELATIONSHIP WITH OTHER DELIVERABLES

This deliverable is directly related to D5.4 “Design and implementation of the Virtual Loom”, where the state of the art in representing textiles in 3D forms was given, the design and implementation of the virtual loom was detailed, and 3D models of textiles were derived as examples, having in consideration different weaving techniques.

This deliverable is also related to D5.6 “Production of 3D printed textiles for museums, educational institutions and creative industries”, as some of the 3D printed models will be derived making use of the 3D printing components here described.

2.3. DESCRIPTION OF THE 3D PRINTING COMPONENTS AND RESULTS

In this section we describe the 3D printing components. As explained above, this functionality is integrated as part of the Virtual Loom (recall Figure 16), from which users can export the generated 3D model as STL files.

In order to build the STL, the 3D printing components of the Virtual Loom take into account the specifications of standard 3D printers, in such as way that the produced STL is printable

in a standard 3D printer. Basically, a simplification of the geometry is needed, because the virtual yarns modelled in the Virtual Loom (Figure 17) have too much resolution to be processed in 3D printing software. In addition, the printers do not print the internal faces of the threads, but simply the surface of each of the visible faces of those yarns.

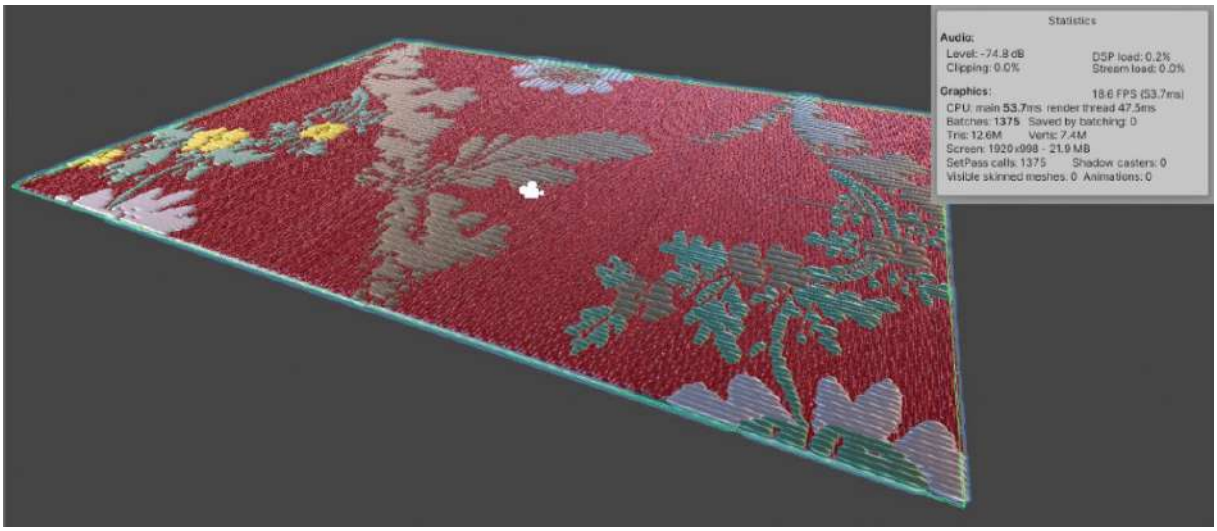


Figure 17. Virtual 3D representation of a textile (espolín technique) as produced with with the Virtual Loom. It contains 12.6 millions of triangles.

Therefore, the solution is to generate the hull mesh of the fabric. This surface does not contain the inner faces of the threads, but simply the faces visible from the front, and the faces visible from the back. The resulting hull mesh has a lower amount of geometry, of the order of 25% less in the tests that we have carried out so far. The most important thing is that it has a topology that is easier to process, since there are no intersections between triangles.

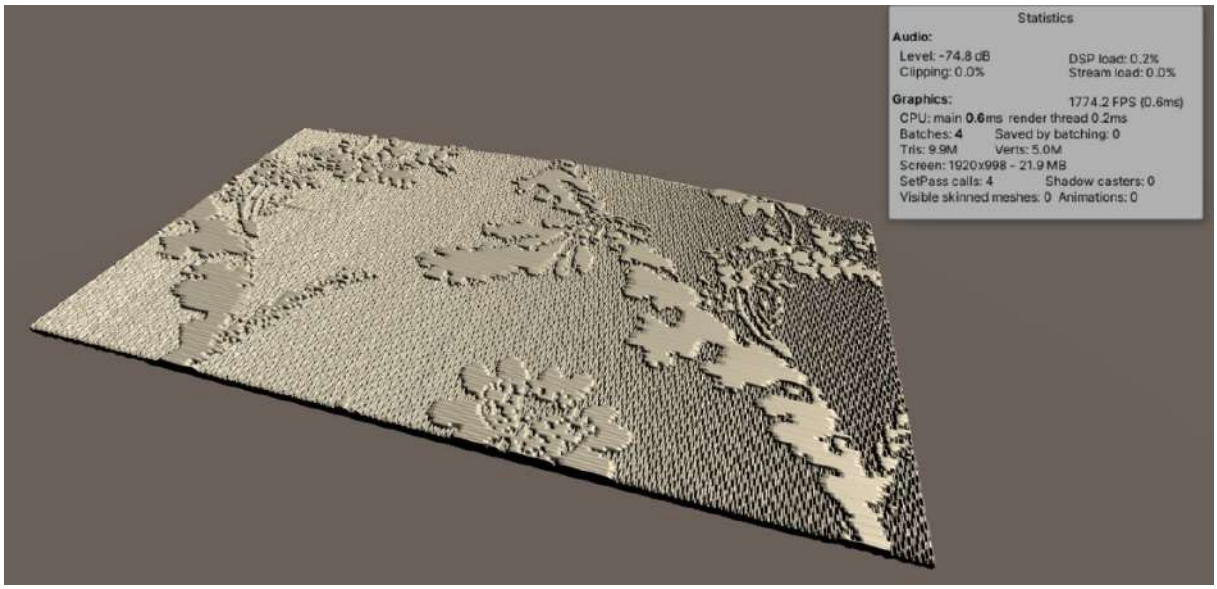


Figure 18. Virtual 3D representation of a textile (espolín technique) as produced with with the 3D printing components implemented in the Virtual Loom. It contains 9.9 millions of triangles (22% less than the original 3D model depicted in Figure 17).

On the contrary, the geometry of the unprocessed yarns (Figure 17) contains a large number of triangles that intersect with each other due to the proximity of the yarns themselves. The intersections of the triangles mean that the calculation of the volume to be printed is much more complex, and in many cases produces an unwanted result. Therefore, by using the hull mesh (Figure 18), the 3D printing software calculates the printing "slices" more quickly and without errors derived from the intersections between triangles, making the 3D printing of modelled fabrics affordable even with high level of detail.

In order to derive the hull mesh, we proceed as follows:

- The height of the yarns is sampled using a virtual camera from the top. This virtual camera is configured with an orthographic projection, coinciding with the size of the fabric, and values of Near and Far according to its thickness (Figure 19).

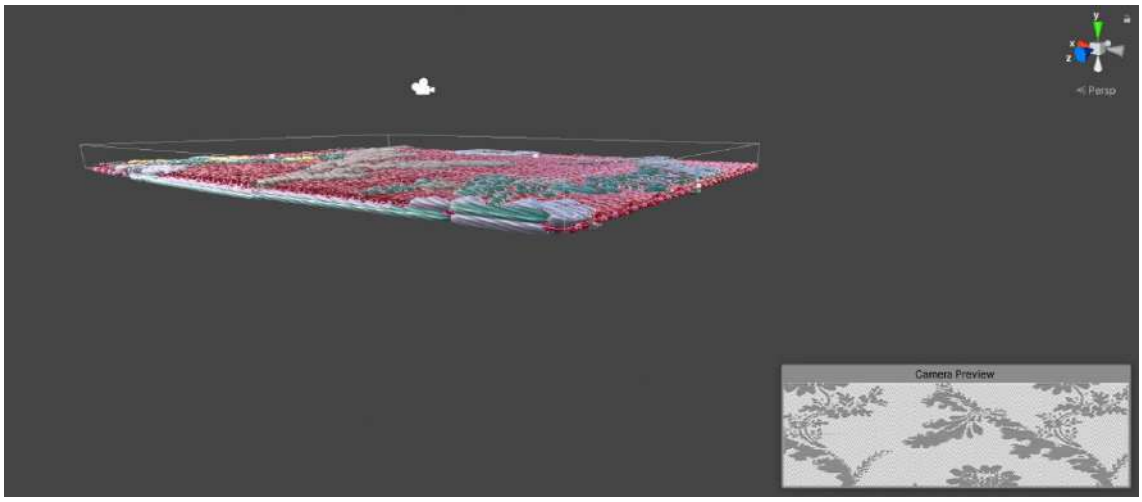


Figure 19. Graphical representation of the orthographic projection configured to sample the height of the face side of the fabric.

- The height is sampled using the Z-Buffer captured by the camera, that is the depth information. The depth information must be processed in a shader to transform it into linear distance, since the Z-Buffer is not linear but has a higher resolution in the areas near the Near plane, and smaller in the areas near the Far plane. As a result, a grayscale image called Height Map is obtained (Figure 20).

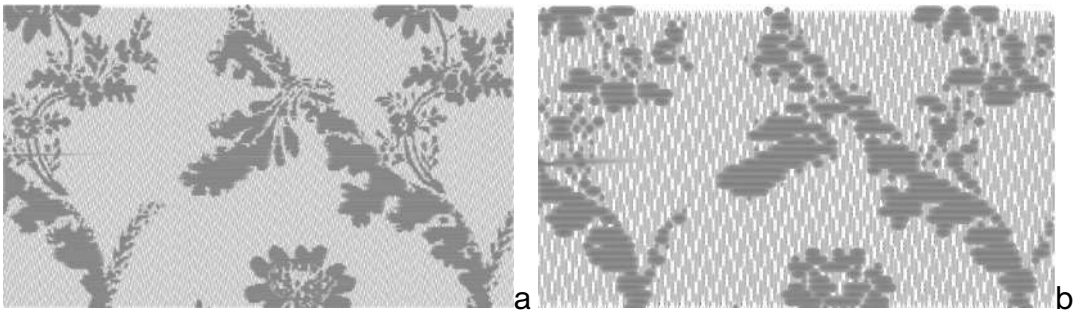
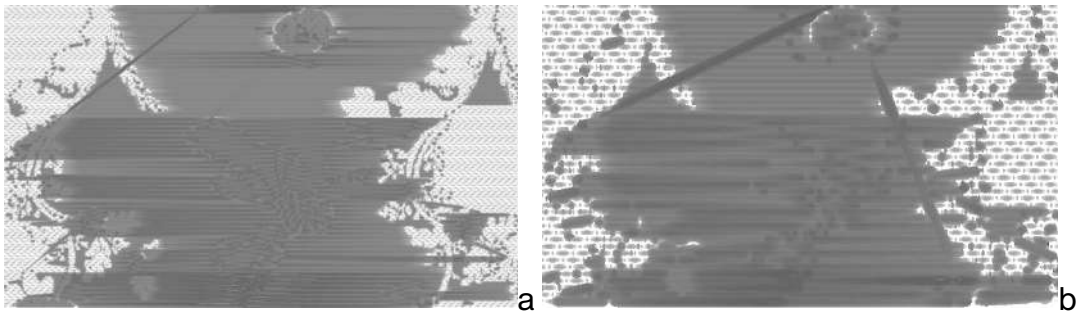


Figure 20. Heights maps of the face side of an espolín in two levels of detail: a) high (450 warp yarns); b) low (150 warp yarns). In the images, the lower gray level (darker) means less distance to the virtual camera, that is, greater height.

- A similar process is done for the reverse side of the textile (Figure 21).



a b

Figure 21: Heights maps of the reverse side of an espolín in two levels of detail: a) high (450 warp yarns); b) low (150 warp yarns). In the images, the lower gray level (darker) means less distance to the virtual camera, that is, greater height.

- Using both height maps, a regular mesh is created for each face of the fabric. This mesh is similar to a "fabric dropped on the 3D model, which fits perfectly to the shape of the yarns."
- Finally, both meshes are joined by the four sides to create a closed volume called hull mesh (recall Figure 18).

The size of the model to be printed is configurable, both in the virtual loom software itself and in the 3D printing software. Specifically within the Virtual Loom, as explained in deliverable D5.4, one can define the level of detail of the fabric. At a higher level of detail, the 3D model of the fabric is formed by a greater number of yarns, and it is of a larger size. Therefore, the “real” dimensions of the STL are given depending on the number of yarns that the model has. To that end, we consider a constant size of the yarn given by its diameter.

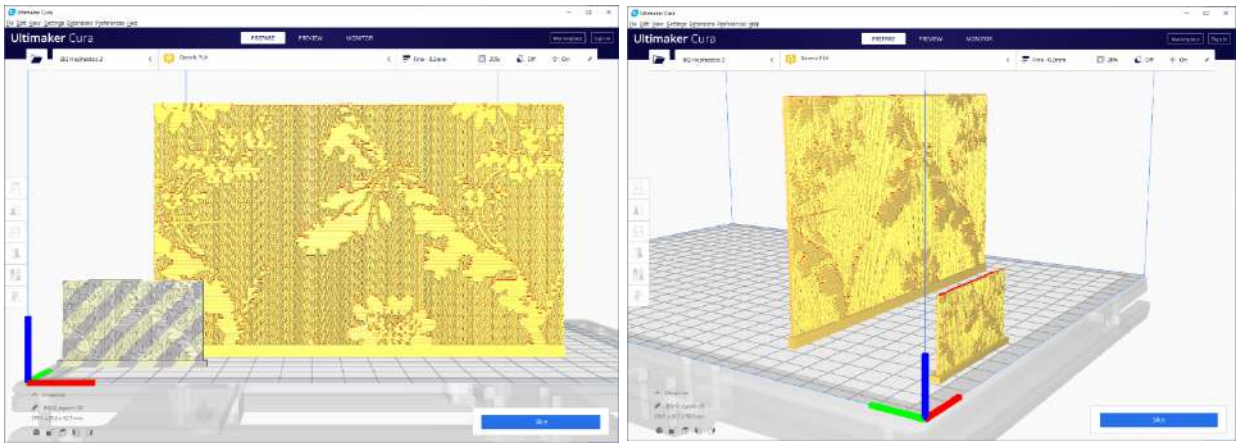


Figure 22. Capture of CURA 3D printing software, where the two models with different sizes (shown in Figure 20 and Figure 21) have been loaded. The one bigger in size, corresponds to the model with the high level of detail.

The example given in this section (the espolín fabric) was printed in a standard 3D printer, making use of the 3D models with low and high level of detail (shown in Figure 20 and Figure 21). The following images in Figure 23 show some results, for the model with high level of detail.

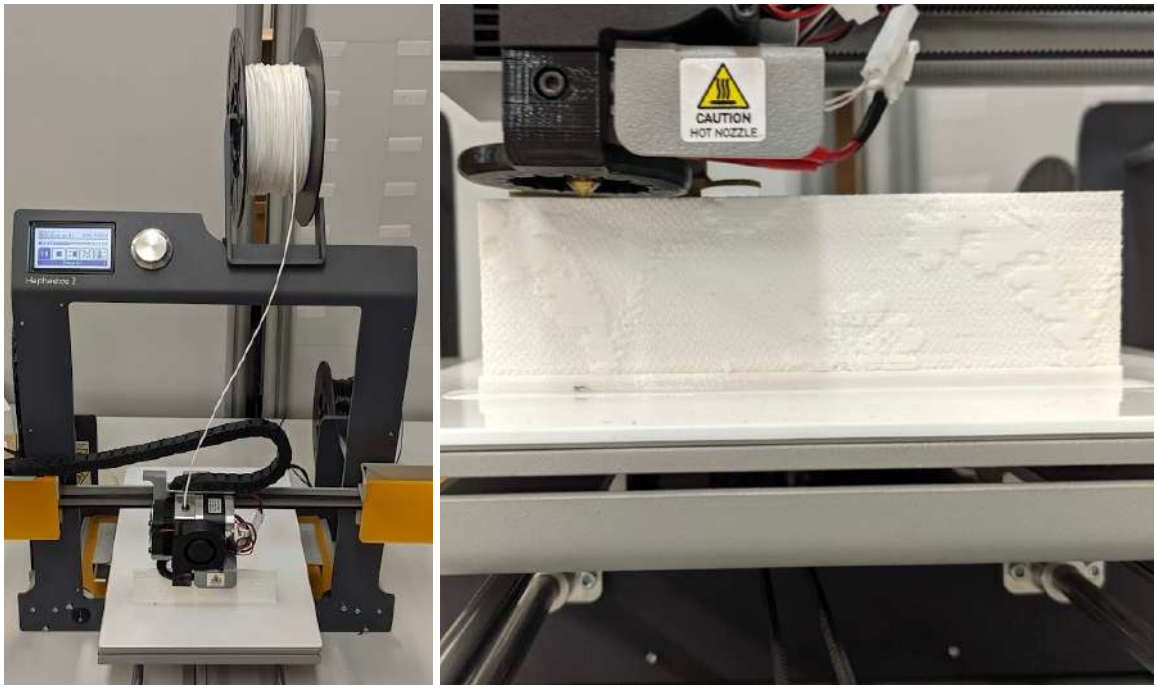


Figure 23. 3D printing of the STL generated with the Virtual Loom, for the example shown in Figure 22 with the high level of detail.

3. CONCLUSIONS

In this document, we have described the details of the design and implementation of the visualization (SILKViz tool) and the 3D printing components.

Regarding the visualization componets, a section of results shows samples of the SILKViz tool with the SILKNOW knowledge graph data. These experiments describe how versatile the SILKViz tool is in producing 3D spatio-temporal visualizations. The SILKViz tool can be downloaded from GitHub; an executable version can also be downloaded from the SILKNOW project web page. The SILKViz tool is currently being tested as part of Task 5.7. All release versions are accessible through http://silknow.eu/silkviz/, and a final version will be delivered at the end of the project as part of deliverable D5.7.

Regarding the 3D printing components, they are integrated as a functionality in the Virtual Loom, which was described in D5.4. The Virtual Loom can be downloaded from GitHub and the release versions can be accessed through http://silknow.eu/virtualLoom/.

4. REFERENCES

1. Kehrer, J., Hauser, H.: Visualization and Visual Analysis of Multifaceted Scientific Data: A Survey. IEEE Transactions on Visualization and Computer Graphics. 19, 495–513 (2013).
2. Liu, S., Cui, W., Wu, Y., Liu, M.: A survey on information visualization: recent advances and challenges. The Visual Computer. 30, 1373–1393 (2014).

3. Lavalle, S., Lesser, E., Shockley, R., S. Hopkins, M., Kruschwitz, N.: Big Data, Analytics and the Path From Insights to Value. (2011).
4. Zhong, C., Wang, T., Zeng, W., Müller Arisona, S.: Spatiotemporal Visualisation: A Survey and Outlook. In: Arisona, S.M., Aschwanden, G., Halatsch, J., and Wonka, P. (eds.) Digital Urban Modeling and Simulation. pp. 299–317. Springer Berlin Heidelberg, Berlin, Heidelberg (2012).
5. Bach, B., Dragicevic, P., Archambault, D., Hurter, C., Carpendale, S.: A Review of Temporal Data Visualizations Based on Space-Time Cube Operations. Eurographics Conference on Visualization (EuroVis 2014). 23–41 (2014).
6. Gruninger, M., Obrst, L., Schneider, T., Fran, Q., Baclawski, K., Bennett, M., Brickley, D., Berg-Cross, G., Hitzler, P., Janowicz, K., Kapp, C., Kutz, O., Lange, C., Levenchuk, A., Rector, A., Spero, S., Thessen, A., Vegetti, M., Vizedom, A., Yim, P.: Ontology Summit 2014 Communique: Semantic Web and Big Data Meet Applied Ontology. (2014).
7. Bennett, M., Baclawski, K.: The role of ontologies in Linked Data, Big Data and Semantic Web applications. Applied Ontology. 12, 189–194 (2017).
8. Baclawski, K., Bennett, M., Berg-Cross, G., Casanave, C., Fritzsche, D., Luciano, J., Schneider, T., Sharma, R., Singer, J., Sowa, J., Sriram, R., Westerinen, A., Whitten, D.: Ontology Summit 2018 Communiqué: Contexts in context. (2018).
9. Konys, A.: Ontology-Based Approaches to Big Data Analytics. In: Kobayashi, S., Piegat, A., Pejaś, J., El Fray, I., and Kacprzyk, J. (eds.) Hard and Soft Computing for Artificial Intelligence, Multimedia and Security. pp. 355–365. Springer International Publishing (2017).
10. D. Dou, H. Wang, H. Liu: Semantic data mining: A survey of ontology-based approaches. In: Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015). pp. 244–251 (2015).
11. Neches, R., Fikes ,R., Finin , T., Gruber, T., Patil, R., Senator ,T., Swartout, R.W.: Enabling Technology for Knowledge Sharing. AI Magazine. 12, 36–56 (1991).
12. D. Guo, Yi Du: A visualization platform for spatio-temporal data: A data intensive computation framework. In: 2015 23rd International Conference on Geoinformatics. pp. 1–6 (2015).
13. Dudáš, M., Lohmann, S., Svátek, V., Pavlov, D.: Ontology visualization methods and tools: a survey of the state of the art. The Knowledge Engineering Review. 33, e10 (2018).
14. Nazemi, K., Burkhardt, D., Ginters, E., Kohlhammer, J.: Semantics Visualization – Definition, Approaches and Challenges. Procedia Computer Science. 75, 75–83 (2015).
15. S. M. Falconer, R. I. Bull, L. Grammel, M. Storey: Creating Visualizations through Ontology Mapping. In: 2009 International Conference on Complex, Intelligent and Software Intensive Systems. pp. 688–693 (2009).
16. Polowinski, J., Voigt, M: VISO: A Shared, Formal Knowledge Base as a Foundation for Semi-automatic InfoVis Systems.
17. Polowinski, J.: Towards RVL: A Declarative Language for Visualizing RDFS/OWL Data. In: Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics. pp. 38:1–38:11. ACM, New York, NY, USA (2013).
18. Peuquet, D.J.: It's About Time: A Conceptual Framework for the Representation of Temporal Dynamics in Geographic Information Systems. Annals of the Association of American Geographers. 84, 441–461 (1994).
19. Andrienko, N., Andrienko, G., Gatalsky, P.: Exploratory spatio-temporal visualization: an analytical review. Journal of Visual Languages & Computing. 14, 503–541 (2003).
20. Ku, W.-Y., Liaw, Y.-P., Huang, J.-Y., Nfor, O.N., Hsu, S.-Y., Ko, P.-C., Lee, W.-C., Chen, C.-J.: An Online Atlas for Exploring Spatio-Temporal Patterns of Cancer Mortality (1972-2011) and Incidence (1995-2008) in Taiwan. Medicine. 95, e3496–e3496 (2016).
21. Hengl, T., Roudier, P., Beaudette, D., Pebesma, E.: plotKML: Scientific Visualization of Spatio-Temporal Data. Journal of Statistical Software. 63, 1–25 (2015).
22. Clauset, A., E J Newman, M., Moore, C.: Finding community structure in very large networks. (2005).

23. Jänicke, S., Heine, C., Scheuermann, G.: GeoTemCo: Comparative Visualization of Geospatial-Temporal Data with Clutter Removal Based on Dynamic Delaunay Triangulations. In: Csurka, G., Kraus, M., Laramee, R.S., Richard, P., and Braz, J. (eds.) Computer Vision, Imaging and Computer Graphics. Theory and Application. pp. 160–175. Springer Berlin Heidelberg (2013).
24. Kraak, M.J.: Timelines, temporal resolution, temporal zoom and time geography. (2005).
25. Lee, Devillers, R., Hoeber, O.: Navigating spatio-temporal data with temporal zoom and pan in a multi-touch environment. (2014).
26. Wang, C., Ma, X., Chen, J.: Ontology-driven data integration and visualization for exploring regional geologic time and paleontological information. Computers & Geosciences. 115, 12–19 (2018).
27. Hewagamage, K., Hirakawa, M., Ichikawa, T.: Interactive Visualization of Spatiotemporal Patterns Using Spirals on a Geographical Map. (1999).
28. Guo, D.: Flow Mapping and Multivariate Visualization of Large Spatial Interaction Data. (2009).
29. Google Maps, https://cloud.google.com/maps-platform/, (2019).
30. Zerdoumi, S., Sabri, A.Q.M., Kamsin, A., Hashem, I.A.T., Gani, A., Hakak, S., Al-garadi, M.A., Chang, V.: Image pattern recognition in big data: taxonomy and open challenges: survey. Multimedia Tools and Applications. 77, 10091–10121 (2018).
31. Zhao, B., Ong, E., Lin, Y., Liu, Y., Xiang, Z., He, Y., Zheng, J., Mungall, C., Courtot, M., Ruttenberg, A.: Ontobee: A linked ontology data server to support ontology term dereferencing, linkage, query and integration. Nucleic Acids Research. 45, D347–D352 (2016).
32. Verhodubs, O.: Realization of Ontology Web Search Engine. (2017).
33. Dudáš, M., Zamazal, O., Svátek, V.: Roadmapping and Navigating in the Ontology Visualization Landscape. In: Janowicz, K., Schlobach, S., Lambrix, P., and Hyvönen, E. (eds.) Knowledge Engineering and Knowledge Management. pp. 137–152. Springer International Publishing (2014).
34. CIDOC Documentation Standards Group: CIDOC Conceptual Reference Model (CRM) – ISO 21127:2006., (2006).
35. Erlangen CRM/OWL. CIDOC-CRM Implementation., http://erlangen-crm.org/, (2013).


VNIVERSITAT DE VALÈNCIA
UNIVERSITÀ DEGLI STUDI DI PALERMO
Leibniz Universität Hannover
MONKEYFAB
Institut "Jožef Stefan" Ljubljana, Slovenija
Instituto Cervantes
cnrs
EURECOM
GARÍN 1820