

Organizing personal web 2.0 content with Hypersea

Editor(s): Name Surname, University, Country

Solicited review(s): Name Surname, University, Country

Open review(s): Name Surname, University, Country

Georgios Styliaras^{a,*} and Sotiris Christodoulou^b

^a *University of Western Greece, 2 Seferi Street, Agrinio 30100, Greece*

^b *University of Patras, Rio, Patra 46500, Greece*

Abstract. Hypersea is a system based on spatial hypertext, node linking and other hypertext research features and its aim is to facilitate users to visually organize content originated from web2.0 and other multimedia sources. This paper discusses Hypersea philosophy and principles, and provides a discrete catalogue of its modelling features. Hypersea can be used for any kind of on-line or off-line content. However its design is focused on information pieces coming from online web2.0 applications, that are stored as structured nodes with visual and spatial characteristics that can be explicitly or implicitly linked. Additionally, we illustrate examples for organizing web2.0 content and, vice versa, we propose how existing web2.0 applications may benefit from Hypersea visualization characteristics. For example, blog entries and Facebook profiles, which are displayed as long text-based pages, can be reorganized in order to enable the user to focus and navigate towards the desired information. Finally, implementation details along with some evaluation results and future work conclude the paper.

Keywords: web 2.0, spatial hypertext, linking, ontology

1. Introduction and Discussion

HyperSea [17] is a system that focuses on the spatial presentation of web 2.0 content and related information spaces. The system denotes and combines previous research results in spatial hypertext and other and sound design methodologies (e.g. Jacob Nielsen's usability rules [12]). For example, spatial hypertext systems [10, 15] enable integrating content from different sources, ambiguity and implied relationships and annotations. Furthermore, these systems employ visual cues massively and they succeed in managing heterogeneous information.

The proposed system focuses on the material embedded in a web 2.0 page as an object (such as a blog entry, a video clip or a song) and not just its URL. It supports importing local or web multimedia resources, part of a web page or whole web pages of any kind but with special focus on web 2.0 applica-

tions' resources. Moreover, Hypersea introduces an enrichable set of metadata as attribute/value pairs based on the objects' origin that should be defined for the various web 2.0 objects and structures. The environment also supports special metadata like date(s) and place for the majority of the nodes. Thus, the information may be placed on maps or sorted chronologically. For example, present YouTube videos on the whiteboard sorted by their upload dates.

The Hypersea environment proposes visual cues, which are drawn from previous research results and can be applied on objects automatically according to their origins, in order to reduce the user's effort for comprehending the presenting material. The set of visual cues is kept as low as possible to avoid overwhelming the user with a lot of choices. HyperSea supports complex content operations by performing simple and homogeneous interface actions and by exploiting the arrangement of content in space. Such

* Corresponding author. E-mail: gstyl@uoi.gr.

operations include creating information structures and then viewing, searching and annotating them.

Conclusively, Hypersea tries to combine the expressiveness of a drawing program for creating, moving and linking content nodes with the structuring capabilities of XML and ontology technologies, where content is organized in fields, tagged and related by identities and supports referential integrity rules on linking and updating. Furthermore, unlike most spatial hypertext systems, the environment supports the use of explicit links among nodes and indirectly some spatial relationship for representing different levels of relations. In HyperSea the nodes' viewable content is changed according to zoom level (for simplicity's sake two levels of zoom are supported). Finally, as in most spatial hypertext systems, the environment supports navigating among critical stages of some web 2.0 content representation, so that it is possible for another user to follow the rationale of the user that evolved the representation.

Although a lot of research has taken place regarding hypertext content and related methodologies, environments, structures, navigation methods and interaction mediums, this research tends to be secluded from recent massively used web 2.0 applications. These applications ignore these research results and provide a primitive text-based interface through which users perform operations, not always effectively. More specifically, regarding research results, spatial hypertext allows the placement and interaction of more than one hypertext nodes on a single area. Link type and the distance of nodes in space may express more information about the linking of nodes.

Similarly, graph visualization allows complex structures to be meaningfully presented as structured sets of nodes. Hypertext patterns define a classification of navigation types among hypertext nodes and allow extracting results about the expressiveness and the quality of hypertext. On a more abstract level, knowledge networks and ontologies permit the representation of knowledge through structured networks of content nodes. On the other hand, adaptive hypermedia systems permit the adjustment of the projected hypertext content according to the user's navigation habits. Finally, hypertext methodologies and tools suggest a structured process for defining, structuring and presenting a network of hypertext content of different link and node types.

As previously stated, despite this previous work, the dominance of HTML and the World Wide Web

has imposed a linear, text-based representation of hypertext, where nodes appear as a text-page enriched with multimedia content and simple, unary hyperlinks that point to other pages or multimedia content. This characteristic has become even more evident in web 2.0 social applications, in most of which complex relations and links are represented either as long pages of text, or are just implied in an indirect way. For example, MySpace is a popular medium for music groups that allows them to present themselves, upload some of their music and keep in touch with their fans. A typical page in MySpace, especially for popular music groups, is a long page that embeds the identity of the group and a music player at the top. Then, fan comments and reviews unfold requiring several page-downs in order to properly view them. Similarly, in Flickr, a user has to perform several clicks and actions in order to view just a set of photos in a linear way. In Facebook, friends, events and requests appear as linear lists. In YouTube, lists of related videos appear next to the projected video along with long lists of comments. In every case, users are not offered an overview of information and it's difficult for them to detect how different kinds of information relate to each other. For example, how friends relate to other friends through events, how group of friends relate to each other, how photos relate to places, events and friends. Furthermore, users can hardly have a clue about historicity of information or how search results relate to the main type of presented information, when performing inquiries. Finally, users can rarely find interlinks between resources of different web2.0 applications.

This is where previous hypertext research can overcome these limitations. Starting from the early years of hypertext research in [11], it is stated that structure discovered in spatial hypertext may be the basis for collaboration or interaction. The need for visualizing tools that represent hypertext structures, not as text links, but in space, is outlined in [1]. The authors in [19] have developed a series of spatial hypertext systems that support early stages of linear-information authoring, such as paper writing and movie making. Alternative mechanisms for figure navigations are presented in [3]. The use of structured hypertext in order to support knowledge representation in the form of ontologies or in documents is analyzed in [4] and [5]. 1001stories [2] is a powerful hypermedia development toolkit that promotes structured content and navigation structures but is limited to generate text-based hypermedia pages. [9] presents an infrastructure for browsing and multimedia blog-

ging of Web-based information anchored with physical places in an urban environment. In [6], the authors have developed a force directed network interface to visualize the result sets provided by a social bookmarking site.

On the other hand, some research has taken place in cooperative hypermedia. DeVise [7] is a Dexter-based cooperative hypermedia system that supports several modes of cooperation on shared hypermedia materials. XCHIPS [18] focuses on synchronous hypermedia cooperation, whereas SEPIA [8] provides support for cooperative writing of hypermedia documents. [13] utilizes component-based cooperative hypermedia to organize shared enterprise workspaces that contain team and process structures, information contents and their corresponding tools. These systems focus mainly on cooperative work but don't take into account web 2.0, spatial properties and visual semantics.

Other recent trends that are related to Hypersea, include web mashups, feeds aggregators and webtops. A web mashup combines content from two or more sources, in order to create new services or added-value information. In news content, making a collection of web feeds accessible in a single location for easy viewing is known as aggregation, which is performed by a web-based aggregator, such as Google Reader or Bloglines. More advanced methods of aggregating feeds are provided via WebTops that are based on AJAX coding techniques and XML components known as Web widgets. WebTops are personalized services which provide a roaming desktop that can host all of user's most common Web information such as emails, news, RSS/Atom feeds, Calendar, etc. Most known WebTops hosting services are Netvibes, PageFlakes, iGoogle, MyYahoo!. However, all these tools are oriented in combining information sources together and especially up-to-date information (like news, weather, emails, etc.) but cannot be used for organizing and sharing information found in the web. Hypersea supports such functionality, as users can create complex spatial structures of information sources and links among them.

It is true that web 2.0 applications have played a tremendous role in adding added value to pre-existing raw content, especially bringing together people with common interests. However, by exploiting previous hypertext research, more expressiveness and navigation capabilities may be added to the same information. In other words, web 2.0 social applications deserve more powerful interfaces and structur-

ing capabilities without sacrificing their simplicity. What's more, various web 2.0 social applications host similar information about users, events and photos, which is not related by any means. There are also web 2.0 applications with specialized content that may be related to social applications. For example, a Facebook user to point to flickr images, or a YouTube movie trailer to point to a Wikipedia movie entry. The above features are embedded in the design of the Hypersea environment, which can act as a reference guide for exploiting hypertext research in web 2.0 applications.

In this paper, after a short presentation of the Hypersea environment, we encode Hypersea's functionality as a hierarchical structured list of self-contained features to be used as a checklist for the presentation and implementation of related web 2.0 applications. Additionally, we present case studies of using the Hypersea environment. On the opposite, we propose theoretical examples of embedding the environment's functionality on existing web 2.0 applications. More specifically, we propose how existing web 2.0 applications could benefit from Hypersea and, in parallel, we point out the limitations of web 2.0 applications. Finally, we outline some implementation details regarding Hypersea and the use of the ontology, and conclude the paper with some evaluation results and future work.

2. Hypersea Environment

HyperSea is an environment for organizing and presenting web 2.0 content. The environment allows a user or many users to organise their information sources in one large space, called *Archipelago*, which can be authored and viewed with two-levels of detail. This decision was made in order to keep the environment simple and inhibit users from creating deep and complex hierarchies of data. Instead, Hypersea encourages users to represent deep hierarchies in space.

In the first level of detail the user sees an archipelago spatially divided into labelled *seas*, which are further divided into *complexes of islands*, each of one comprises individual *islands*. In the first level of detail, users cannot view the information within an island (unless the archipelago's space is small, in which case they can).

In the second level of detail the information within islands (nodes, annotations, structs of nodes and links among them) is visible. In this level of de-

tail all explicit links among nodes and structs belonging to different islands are also visible. These links are also visible in the first level of detail as links between the islands but the end points of the link are placed inside the islands symbol.

Essential Definitions

Node: an abstraction of a simple piece of information within information space. Based on the analysis made in the previous section, a node may represent (we refer to it as *content type*): a user, a group of users, a web 2.0 information object (according to the kind of object different metadata will be attached to the node), a normal web resource not belonging to a web 2.0 application (a web page, image, video, document, etc.), a file in user's hard disk which is uploaded into the environment. Regarding users and groups, a user node allows access to user's content that has been dragged to the environment from several web 2.0 applications. A Facebook or MySpace account of the same person is mapped to the same user node. Users belonging to the same group may share each other's content that has been dragged in the environment. In this way, users of the same group agree to share diverse content, as in an existing web 2.0 application.

Struct: A spatial structure that incorporates many nodes organized in a kind of structure, such as [14]:

Composite: a structure consisting of a regular spatial arrangement of nodes with different types. Used to denote implicit semantic relationships among different types of nodes.

List: an ordered or unordered list of nodes of the same type. A list can be presented ordered by any metadata selected by the author during authoring or another user during viewing. Moreover, a list can be presented as a *stack* if there is not enough space or the list is too long. Lists are commonly used by people to organize material in categories or sets.

Collection: a set of nodes of different types. Can be also presented as a **heap**. Used when collecting information but not yet sure how to organize it.

Map: a set of nodes placed over a background image that can be a map, an aerial photo, a diagram, a drawing, an organogram of a company, etc.

Numerous combinations of these structs are possible in order to choose the best one for the structuring of your information space. Such combinations are a list of lists (nested lists), a list of composites, a composite including a node and a list, a collection including a list and a composite, a list of maps, etc.

Annotation: It is a special object within our environment. It is not a node in the sense that it cannot be linked, but it is directly associated with the node,

struct or island placed near or over it. It usually represents comments, messages to other users, reviews, ratings of users, etc.

Island: An island is a subspace that usually reflects semantic categorization of the whole information space. It may include numerous nodes, annotations and structs of any kind. It may include internal or external links. Islands may not include other islands for the sake of simplicity. Usually, a collection of information resources for a thematic subject or a user are represented as an island but also as a struct or a simple node, depending on the volume and complexity of information material and the task the environment user wants to accomplish.

Archipelago: The whole information space is called Archipelago, meaning a large cluster of islands. Deep hierarchies of islands are presented in space with labelled areas (**seas**) and subareas (**complexes of islands**, i.e. structs of islands). However, individual nodes or structs may also be presented inside an archipelago or a sea or a complex of islands.

Link: The link is used to express the relationship between two objects (nodes, structs or islands). This relation in space is expressed either implicitly (via proximity in space, participation in the same struct, same visual characteristics like color) or explicitly via typed links (i.e. uni or bi-directional arrows with labels). Explicit links are used to express relations out of the main hierarchical spatial structure of information. Implicit links are also used to express ambiguous relations, where users are not quite sure if they have to relate two items, especially during the task of analyzing some collected information.

Metadata: Any kind of object within HyperSea, including nodes, annotations, structs, islands, links and even the whole Archipelago has its own metadata set that describes its properties. Each node type has an appropriate metadata set according to its web 2.0 kind (the kind of the source web 2.0 application) and content type. For each object type there are some *basic metadata* that always have to be included and some other *optional metadata* that are just suggested by the environment. Moreover, users are able to add their *own metadata*. Some metadata are inserted and managed totally by the environment, like the timestamp of the creation time of the object, the access frequency of the object, etc.

Basic metadata: Title, URL, Description (in XHTML format), Date of Creation, Date of Upload, Place, web 2.0 Object Kind [17], Content Type (Document, Image, Video, Streaming Video, Audio, Streaming Audio, Web page, User, Community)

Optional metadata: depending on the kind and type of object the environment suggest some optional metadata for user to include.

Technical metadata: Automatic technical properties defined for a node, such as the size, width, height and color depth for a picture or the duration and encoding of a sound or video file.

User-defined metadata: user can add any number of metadata

Date, place, kind, type of object, creation time, and any other metadata can be used to search for nodes or filter nodes during viewing of the Archipelago.

For example a film trailer about Normandy invasion created in 1962, uploaded to YouTube in 20th December 2006 has the following attributes, any of which are suffice in order for a node to be defined in a valid way:

Basic Attributes:

URL:

http://www.youtube.com/watch?v=nqFn_pM5QxU

TITLE: The Longest Day

COMMENT: an old movie

CREATION DATE: 1962

UPLOADED DATE: 20th December 2006

PLACE: Normandy, France

WEB 2.0 KIND: YouTube video

CONTENT TYPE: Streaming Video

Optional Attributes:

CATEGORY: Entertainment

Technical Metadata:

LENGTH: 3:06 min

WIDTH: 425

HEIGHT: 344

RESOLUTION: 360p

VIEWS: 131,414

Levels of comprehension

A spatial interface allows users to take advantage of their considerable visual memory and pattern recognition [16]. Jacob Nielsen wrote in 2005 [12] that "Once we get screens the size and resolution of a broadsheet newspaper, the user interface will change. It will become possible to rely more on spatial hyper-text and less on linear scrolling. In fact, the very concept of a page may vanish and be replaced by higher-level aggregate units that combine multiple data feeds." Information spaces produced by Hypersea environment provide users with alternative ways of content comprehension within this space and its relations. Levels of comprehension are:

- 1st level (archipelago): the position, the size, the color and the labels of the islands, structs or nodes can provide information just at a glance. For instance the size of an island indicates the volume of content inside it.
- 2nd level (island): within each island: the position, the size, the color and the label of structs, or nodes.
- 3rd level (structs): grouping / order of items under a label
- 4th level (nodes and annotations):
 - 1st level categorization of nodes: the color of a node by default indicates its content type. Author or viewer may choose to indicate other metadata. Other visual cues are also used, e.g for a user a small picture in the upper left corner. An annotation is presented as a yellow no-bordered square.
 - 2nd level categorization of nodes: the border color of a node indicates another metadata that user can choose. The default color is black.
- 5th level: other issues like *importance or value* of an island/struct/node can be indicated by the width of borders or the style of text within box (e.g bold, italics, font-size, etc.).

3. Environment Features

In this paper, we aim to define a detailed list of features for Hypersea, carefully selected to meet the requirements of web2.0 users. The list of features is based on research and design methodologies as previously presented. They focus on providing simplicity while performing powerful operations and they support the efficient exploitation of the user's space. This list can guide the design and implementation of other applications that would like to incorporate all or part of the Hypersea's features.

More specifically, we focus on the idea of structured nodes that can be drawn on a whiteboard and are related together either explicitly (via linking them) or implicitly (via proximity or visual cues like color). Nodes have a common appearance with a header, attributes and access to their underlying content, so as not to aggravate the user's learning effort. Node operations, such as select, drag and linking result in respective change in their status, e.g. drag a line from one node to another results in creating a link among the nodes. We engage the absence of

menus and buttons, in order to allow for more natural operations (e.g. group some nodes by encircling them, or delete a node by dragging it out of the whiteboard). The nodes' placement and use of visual cues are also very important design rules. Finally, attention has been given for providing status messages and allowing moving seamlessly among various environment states. More analysis about the environment's design rationale can be found at [17].

The viewer may operate the environment in two distinct modes (editing mode and navigation mode). For every operation mode, the catalogue below enumerates the attributes, operations and interface issues, which suffice to cover the entire environment's functionality, such as the creation and the interaction with nodes, and more complex structs and islands.

Editing mode

Node

1. Creation
 - a. Drag page on whiteboard from web browser or file explorer
 - b. Create node on drop point
2. Appearance
 - a. Box having the origin at top (e.g. MySpace, YouTube, Facebook, Disk etc).
 - b. All nodes have the same size
 - c. Node color: indicates the type of its content. (e.g. sound, video, text, user, web page etc)
 - d. Node border: indicates its value.
 - e. Grid-based placement
3. Operations
 - a. Single click: selects node and shows all its identity and metadata.
 - b. Double click: open content represented by the node (a web resource or a file).
 - c. Delete node (del button): the selected node, all related links and associated metadata are removed
 - d. Copy node: A copy of the node is created with same content and attributes
 - e. New instance: Same node at a different place. Any change on one of node instances affects all instances

Node attributes

1. On the area below the title, there are attributes extracted automatically (whenever possible) from the web page.
2. The user can edit attribute values and insert new attributes and respective values

3. Description can be an XHTML text, so that nodes are able to show formatted content and images.
4. Metadata editing of an object is initiated by double clicking upon it.
5. Addition of new metadata is performed by simply clicking under the last metadata.
6. Metadata properties are structured as Xxxx: xxxxx and grow according to their length.
7. Deleting a metadata is performed by simply dragging it outside the popup window of the object.

Struct

1. Creation
 - a. Draw a rectangular around a set of nodes
 - b. A node may join a struct by simply dragging it inside the struct.
 - c. A node may leave a struct by simply dragging it out of the struct.
2. Appearance
 - a. A rectangular with a header with the title of the struct
 - b. A struct is named and characterized through a set of metadata, as the single nodes.
 - c. In a *map struct* the main area of the rectangular is filled with an image (usually a map) and nodes are placed on this map.
 - d. In a *list struct* the user may specify a metadata to sort items with.
 - e. Unorganized material that is gathered before importing it in the whiteboard may be expressed by a *heap*.
 - f. A stack or an ordered list can represent nodes of the same type like dated blog archives or a set of pictures.
3. Operations
 - a. User may move a whole struct in space by dragging it through its title area.

Annotations

1. Creation
 - a. Double click on an empty area of whiteboard
 - b. Write content. Annotation node size changes as the user continues writing text
2. Appearance
 - a. Yellow squares with no borders.

(1) Islands

1. Kinds

- a. Islands for thematic subjects of information items
- b. Islands for user or users
- 2. Appearance
 - a. They are not colored
 - b. A user island is presented as a rectangular with a small picture of her (or an icon if picture is not provided) at the upper left corner and her name at the upper right corner.
 - c. A user island can be used to associate users of different web 2.0 applications with the same live person; e.g. associate a MySpace user to a YouTube user if it is the same person.
 - d. Group of users are represented either as a complex of islands or as an island.
 - e. A subject island is presented as a rectangular with thick borders and a header with the title of the subject.
 - f. While users view the archipelago in the first level of detail, subject islands are represented smaller but the relative size and position in space remains the same.
- 3. Attributes
 - a. An island is named and characterized through a set of metadata, as the single nodes.
- 4. Operations
 - a. Nodes may join or leave an island by simply dragging them in or out of an island.
 - b. Users may move a whole island in space
 - c. An island may be placed within a struct of islands.

Implicit Links

1. Objects placed near each other are implicitly related as in spatial hypertext systems.
2. Grid placement of nodes is exploited in order to be able to calculate the distance among nodes and define a proximity criterion in searches e.g. find nodes of distance 3 grid squares.

Explicit Links

2. Creation
 - a. Drag line from the first node's title area to the destination node's title area.
 - b. Double-click the origin of a link to convert it to bi-directional
3. Appearance
 - a. Explicit links are represented as labeled uni- or bi-directional arrows.
 - b. An arrow appears in the middle of both nodes' title area

- c. Both ends of the arrow follow the movements of nodes
- d. In the first level of details the links among islands are shown. If there are many links between the objects of two islands, only one link is presented in this level.
- e. In the second level of detail show all explicit links among nodes and structs belonging to the same or different islands
- 4. Operations
 - a. Drag a link arrow away to remove it
 - b. Click in the middle of an arrow in order to set a label about the link that grows automatically according to its length

General Operations

1. Creating a new Archipelago. A name for the Archipelago is required that has to be unique for the current user. The Archipelago is saved (either locally or in a server according to the version of Hypersea) as an XML file with a filename that denotes the user unique ID, the name of the Archipelago and the timestamp of the creation time.
2. Sharing an Archipelago. The creator of the Archipelago can provide a list of others users that can edit and/or view the Archipelago.
3. Saving an Archipelago. Each time the user press the save icon a new version of Archipelago is saved (with the current timestamp). Thus, it is possible in the navigation mode, another user to navigate through the various saved versions of the Archipelago and observe how it was constructed.
4. Loading an Archipelago. The creator or an editor can load the current version of an Archipelago or a previous version of it.

Navigation Mode

Operations

1. Change detail level. If the user moves the mouse pointer over an edge of the visible space, or drag the mouse over a white space, the space is scrolled in this direction.
2. Examine node content
 - a. Select node: When selecting a node, four square dots appear at its corners. In case of multiple node copies, if one node instance is selected then all nodes' instances are selected. If a node instance is not visible at a certain zoom level, a visual indicator ap-

- pears in the edge of the whiteboard pointing the direction of the other node instance.
 - b. View full metadata
 - c. Play multimedia content or load web 2.0 page.
- 3. Traverse links
- 4. Add, Edit, Delete annotations
- 5. Move among the stored versions of an Archipelago. While navigating through old versions, if the user is zoomed in an island and press the next version button (→) or the previous (←) then the corresponding version of Archipelago is loaded and zooms automatically to this island.
- 6. Filter nodes based on metadata. Filtered nodes appeared colored differently

Interface

1. There are just two arrows ← (backward) and → (forward) for moving among the stored versions of the Archipelago or island.
2. There is a question mark (?) for supporting searching
3. There are no menus, popups or lists for supporting the above features.

4. Hypersea Application in Web2.0

The expressiveness of the Hypersea environment may be applied in various web 2.0 applications in order to provide a uniform content base, connect scattered but similar information and provide more supervisory modes of presentation than the ones currently supported. More specifically, the environment may aggregate the different user bases deployed in the various web 2.0 applications. The same person may have registered as a different user in Facebook, MySpace and Picassa or maintain a blog. In this environment, different users associated with the same person are concentrated in the same node. Similarly, various media files are classified under broad categories of type (image, video, sound, text) and kind (song, videoclip, photo). In this context, photos from Facebook, Picassa and flickr are handled in the same way. This aggregation may support chains of information about content of different type and kind. For example, a user may import songs, videos and photos regarding a specific movie. The identity of the movie may be found in imdb, the songs in iTunes, photos

from flickr and the movie's trailer in YouTube. By placing and connecting this information in the environment, the user may create a small island of content regarding the movie and retain links to multimedia files and web 2.0 applications pages of the original content. This island may be enriched later with nodes of Facebook's users that all went to see the movie and flickr photos of the cinema. The same users may be related in some other time as they may attend a concert of a group (taken from MySpace). The environment may be exploited in the same way in order to represent a digital memory of a sequential event (vacations) through photos (e.g. imported from flickr) and places (e.g. imported from Google Earth). Consequently, the environment employs a spatial, node-based platform for presenting related content in web 2.0 applications, by extending limited graph view capabilities as they exist in Facebook for example.

A simple case study based on an earlier implementation deals with the association of a picture to a blog about Archaeology. The blog is entitled Archaeology and the user starts by importing 4 pictures. Then the user draws two nodes and tries to unite them via a line starting from the left node. The result is the creation of two nodes linked by a left-to-right arrow. Clicking on the third picture, this picture gets highlighted by a rectangle and three metadata are inserted: the picture's width and height and a user entered caption. These metadata stay attached to the picture regardless of its position and its appearance. The user chooses to drag the third picture over the first node and attaches in this way the picture to the node. A thumbnail of the picture appears in the node, which when clicked will display the picture's metadata. Simultaneously, a blog link is imported, which gets dragged over the second node and attached to it. A "sign" denotes that the second node refers to a blog link and details about it appear when the node is clicked. Finally, clicking on the arrow, a text about the relation between the nodes is entered. Overall, the resulting structure shows that the picture on the left is presented in the blog attached to the right node. The above editing process is illustrated in Figure 1 as a scheme.

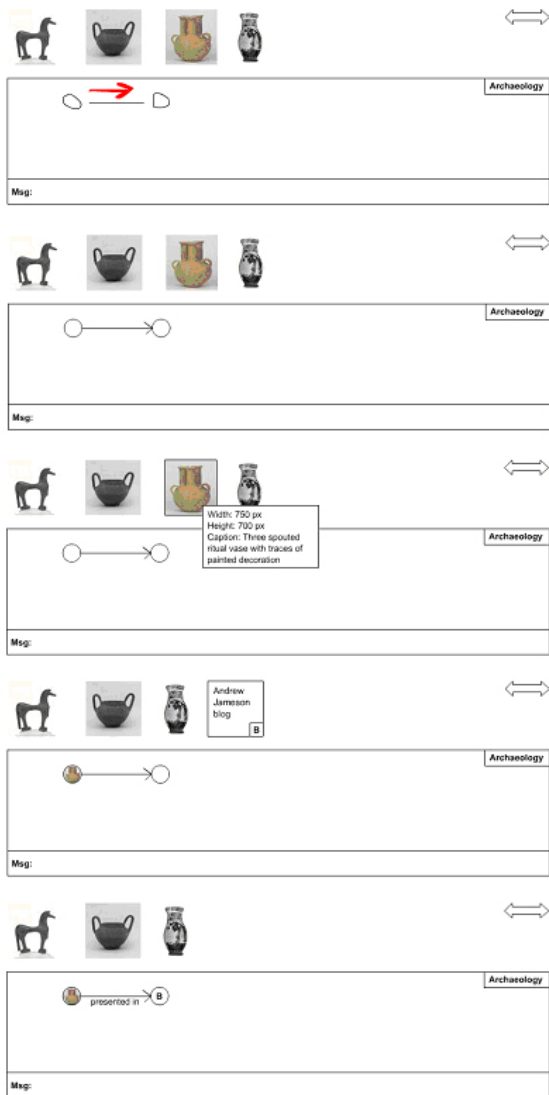


Figure 1. Editing process

Similarly, consider the following scenario of a user's navigation in web 2.0 about the film "Il Divo". The user finds the film in imdb (<http://www.imdb.com/title/tt1023490/>) and downloads the film's poster. (S)he also tries to find the movie's trailer in YouTube by typing "Il Divo". However, owing to the unstructured type of information in YouTube, as in most web 2.0 applications, a bunch of video clips appears mostly dealing with the music group "Il Divo", which is popular in the US. At last, the user manages to find a trailer of the movie at <http://www.youtube.com/watch?v=qnFnHhD-AbY>.

(S)he enjoyed the music score heard in the trailer and finds out from imdb that Teho Teardo is the composer of the film's soundtrack. Searching for him in MySpace, (s)he does not find a page about him – unstructured information strikes again! – but rather finds a group at <http://profile.myspace.com/index.cfm?fuseaction=user.viewProfile&friendID=101900078>, titled BiP_HOp, the music of which is inspired from the film's composer. Finally, the user buys a song of the group from iTunes. By representing this navigation path in an island, all this information about people and content gets structured and remains in this way, while it is charted on a single area. More specifically, as shown in the upper part of the next figure the user has to import the links to imdb, YouTube and MySpace along with the picture displaying the film's poster and the song downloaded from iTunes. By dragging this information in the island appropriately, the user finally gets to represent the information resources and their relationships as shown in the lower part of the next figure. Obviously, more links may be added or further specialized. For example, one could add a node among the imdb and MySpace nodes representing the Facebook page of the composer. The above example of organising related information resources located in web 2.0 applications is illustrated in Figure 2 as a scheme.

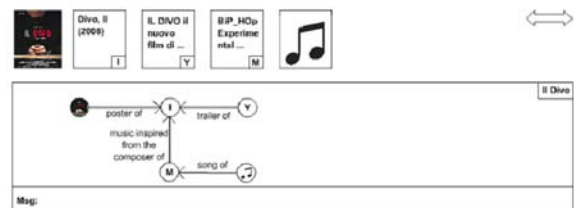


Figure 2. Example of organising related information resources located in web 2.0 applications

Reversely, web 2.0 applications may benefit from such an environment by embedding in their interfaces the capability to structure and distribute their content in nodes that they link to other nodes of the same application or other applications. In this way, the Hypersea environment may act as a plug-in for these applications. For example, an actress having a MySpace page may organize her films as nodes and provide links to imdb. On the other hand, small navigation menus may group navigation links and present them in a more limited area, which does not require

from users to scroll up and down as happens for the archives of blogs.

In order to better illustrate this idea, we present a theoretical example of how a typical Facebook page could be enhanced with the use of the environment. In such a page, there are several navigation options at the top, the middle and the bottom of the page expressed via text links, icons, and tabs (1). There is an extra, stable navigation zone in the bottom with unnamed navigation buttons that repeat some of the navigation options that lay above (2). Online status is also displayed in this zone (3). The News Feed tab extends the visible area of the page and then the user loses the rest of navigation options that lay on the top (4). Similarly, the Friends page (5) lists all accepted friends in a linear list that requires a lot of scrolling in order to be viewed, by losing, in the same time, the overall structure of the page. Requests, applications and inbox messages are handled in the same way. By looking at the page overall, there is a lot of blank space on the left, on the right and in the middle of the page that is not exploited.

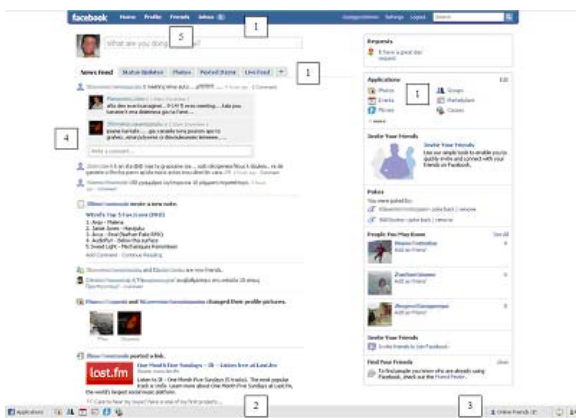
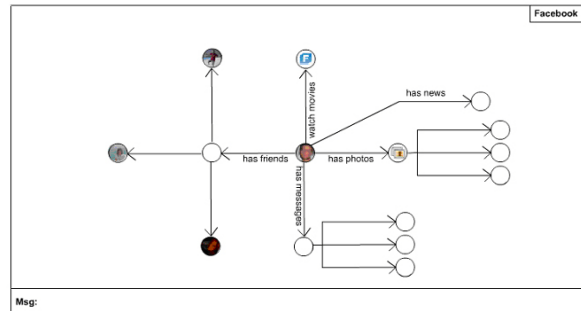


Figure 3. Typical Facebook page

The environment presented does not intend to compete with Facebook's structure but could complement it by offering an alternative way to organize and project the available navigation options as relations in space. For example, unfold most popular

options as nodes near a central node representing the person's page. In this way, distanced nodes, not frequently used would not interfere with frequent ac-



tions. This is a sort of adaptive navigation comparing to the fixed interface offered by Facebook. In the following figure, popular navigation options include a list of selected friends, inbox messages, the applications of Movies and Photos and the top of news feed.

Figure 4. Alternative Facebook

On the other hand, a popular music blog page looks like as in Figure 5. There is a central title (1), links to music events (2), the blogger's identity (3), blog entries as in (4), links to proposed records and books (5), proposed hyperlinks (6), the blog's archive (7) and a music player (8). We had to cut the blog's page, which is once again based on a linear, text-based approach, in two parts (A, B), as the page is very long. It is obvious that a user has to scroll several times up and down in order to locate the right information, for example, when moving among blog entries and the archive or the music player. Alternatively, all this information could be structured around a central node pointing to blog entries and the rest of the information. More frequently visited nodes and new blog entries, which are the blog's essence, will appear bigger and nearer the central node, so the user will focus on the desired information, without having to scroll or losing the blog's overview structure. A reorganization of the bog page shown previously is shown in Figure 6.



Figure 5. Popular blog page

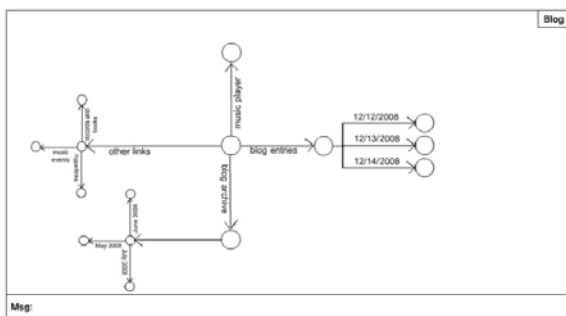


Figure 6. Alternative blog

Consequently, the environment could serve as a plug-in for web2.0 applications in order to provide an overview of their content and a linking mechanism to other applications.

5. Implementation Issues and Architecture

Hypersea has been actually implemented in Adobe Flash with Adobe AIR extensions by using an OWL-based ontology (http://w3.org) in the background for storing the objects' details (e.g. nodes, structs, meta-data, links, etc.). Protégé 4.1 has been used for modelling the ontology. The ontology scheme is shown in Figure 7. Most of the environment's features have been implemented by using scripting that exploits vector graphics positioning, positioning and sizing of images and ontology access routines.

During authoring mode, an ontology is used for storing the objects of the environment. More specifically, textual content and linking with external files are stored in the ontology whereas every node and set is assigned a unique identity. Connections and set members are also stored in the ontology by exploiting their identities. Users see this information only visually during the authoring mode. Identities help also apply integrity rules regarding the content's usage, e.g. when deleting a node, all its relations are also deleted.

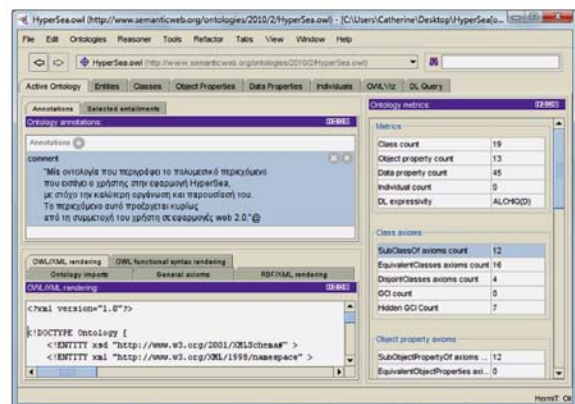
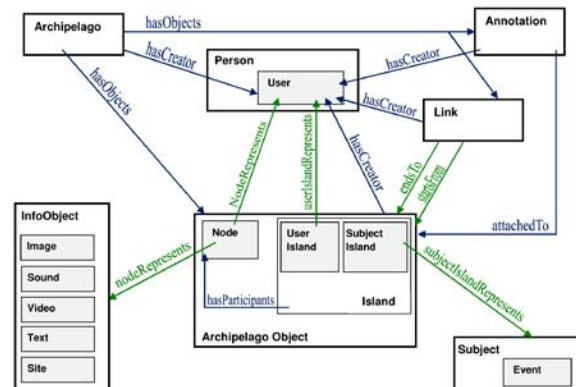


Figure 7. Ontology design

The ontology is saved in XML files. The ontology reflects the environment's status at a certain time. During authoring time, a user may load the environment's status via the ontology that is stored in XML, make changes and save it.

By exploiting the nodes' position on the screens, resizing routines and simple mouse events, it was possible to build the authoring mode of the environment and implement all features that require absolute positioning, such as organizing and arranging nodes and sets; placing nodes in a linear way; playback the island's creation process; creating sets of nodes; importing content in the environment; associating metadata and deleting nodes and sets. After editing, settings and effects preferences are stored in data structures and, finally, in the ontology, when the island representation is saved.

Multimedia files and web pages that are represented by nodes are just linked to the ontology. More specifically, a code is stored in the ontology that refers to the multimedia file's storage path in the file system or the URL of the webpage. By exploiting this code, the environment can generate the multimedia file's actual storage path and present it. In this way, the ontology remains independent from actual multimedia content. Furthermore, external editors may be used in order to edit the files independently of the environment. At any time, a broken-link avoidance mechanism ensures that files and pages are properly linked to the nodes. If a multimedia file is erased or moved, the mechanism will report it and will prompt for a file replacement.

In playback mode, the appropriate content is retrieved from the ontology and embedded in the playback file. The content includes multimedia files, web pages and details about placement. These parameters allow the authoring mode to construct the Dynamic HTML page that projects the memory.

The folder corresponding to a Hypersea instance, as is, may be viewed locally by a web browser or may be transferred to a web server. Furthermore, only in the web-server edition, may the annotation feature be exploited by other users.

Two sample videos that illustrate Hypersea's interaction in real time can be downloaded at <http://www.culture.uoi.gr/proswpiko/gstul/hypersea.zip>. In the first video, there is a user node (gstyl) and a mp3 file that has been already positioned on the surface. The user located the song's video clip in YouTube and drags the video-clip in the environment. It

is clear how nodes from different origins have different colors. Then, the user places the YouTube node in an island, while dragging the mp3 file into the user's island. A link is created among the user's and YouTube's island. It is also illustrated how nodes and links are created and deleted and how islands can be moved, how nodes can be replicated and deleted and, finally, how multimedia files can be activated and how user's actions can be undone. The second video shows how two nodes may be inserted in an island. Both videos show how the status bar reflects the available actions. Figure 8 shows a screenshot while interacting with Hypersea for linking content related to the film Avatar.

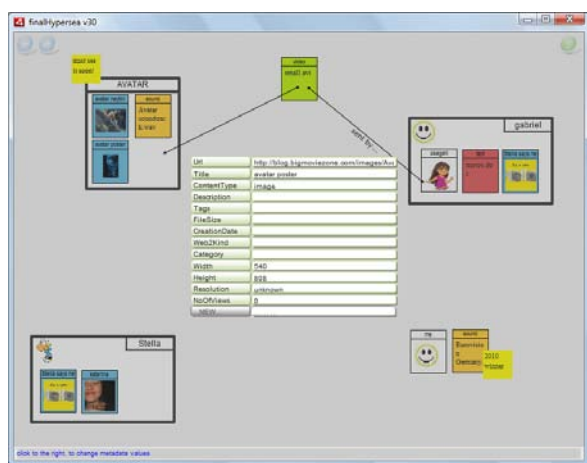


Figure 8. Hypersea environment screenshot

Overall, the environment is based on a simple architecture. It offers three ways to interact: importing multimedia content and web 2.0 page links through the import module; sketching the interaction among content through the design module; and viewing the content by exploiting its structure by the playback module. Finally, actions of all modules end up as relating content nodes enriched with metadata through links and sets. These relationships are mapped in an ontology through the interface translator. Thus, the ontology holds all content and its interconnections and ensures that all information remains consistent despite user actions including link creations and deletions, nodes movement etc.

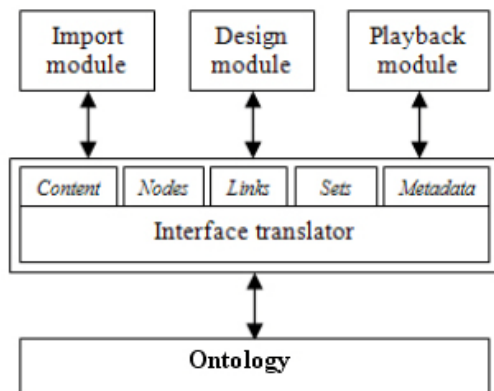


Figure 9. Architecture

6. Environment use and evaluation

The environment has been used in the implementation of the new web site of the Archaeological Museum of Ioannina in Greece (<http://www.amio.gr>). Archaeologists and museum curators had the chance to organize the content of artifacts, collections and museum exhibition areas by using the reasoning of the environment. More specifically, they used the environment in order to gather the multimedia material they had in various web 2.0 applications for monuments of Epirus. A map of Epirus was used in order to place the monuments. Photographs and comments for monuments as well as orderings and time-based and subject-based groupings of the material have been made possible with the environment. A group has been defined, to which all museum's personnel responsible for the web site had enrolled. Users have used islands in order to group artifacts and monuments of different areas. They have also used the facility to import multimedia material and relate it with the islands' nodes. Based on some initial heuristic evaluation regarding the environment, the following results arose:

- Users liked the visual representation of content relationships.
- Users favoured the dragging and replication of nodes in the environment, which resulted in the respective movement of the underlying information.
- Users exploited the capability to reuse the same content nodes in various groupings and orderings in the environment.
- Some time was necessary to get used to the zoom feature, which hid and appeared nodes' de-

tails. Users tended to believe that during zoom out, content was deleted.

- Natural interface actions, such as deleting a node by dragging it out of the environment and the absence of traditional menus were appealing.
- Users liked the sketch-like pause and resume feature of the environment, as they could save the current status and continue working at a later time.
- The view of a web page and the playback of a multimedia file from the environment resulted to be a necessary feature.
- The search interface is complicated and oriented to users with algebraic knowledge.
- Users felt uncomfortable with the notion of user node, as they felt giving away too much personal information, although only information dragged in the environment was shared by every user.
- The freezing of the environment's status was interesting as users could rearrange non destructively the content's layout.

7. Conclusions / Future Work

This paper presented an environment for importing, linking and organizing web 2.0 content and then presenting and further disseminating it. The environment exploits previous hypertext research results and helps organising (on a single whiteboard) content from diverse sources by requiring simple operations from users. The same interface is employed for importing content, editing, linking, presenting and searching. The environment has been implemented in Adobe Flash by using OWL-based ontology in the background for storing the objects' details. The same environment can be used for enhancing the navigation in existing web 2.0 applications. We have also presented the environment's functionality as a list of features, from which a subset can be drawn in order to satisfy a certain usage scenario. In the future, in addition to the initial evaluation results, a complete and large-scale evaluation is needed in order to implement a more functional version of the environment. Finally, more work has to be done in order to propose standard ways to embed Hypersea as a plugin in popular web 2.0 applications, such as Blogspot, Facebook and MySpace.

References

- [1] Bernstein, M. 1998. Patterns of hypertext. In Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia : Links, Objects, Time and Space---Structure in Hypermedia Systems: Links, Objects, Time and Space---Structure in Hypermedia Systems (Pittsburgh, Pennsylvania, United States, June 20 - 24, 1998). HYPERTEXT '98. ACM, New York, NY, 21-29.
- [2] Bolchini, D., Garzotto, F., and Paolini, P. 2008. Investigating success factors for hypermedia development tools. In Proceedings of the Nineteenth ACM Conference on Hypertext and Hypermedia (Pittsburgh, PA, USA, June 19 - 21, 2008). HT '08. ACM, New York, NY, 187-192.
- [3] Buchanan, G. and Owen, T. 2008. Improving navigation interaction in digital documents. In Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital Libraries (Pittsburgh PA, PA, USA, June 16 - 20, 2008). JCDL '08. ACM, New York, NY, 389-392.
- [4] Carr, L., Hall, W., Bechhofer, S., and Goble, C. 2001. Conceptual linking: ontology-based open hypermedia. In Proceedings of the 10th international Conference on World Wide Web (Hong Kong, Hong Kong, May 01 - 05, 2001). WWW '01. ACM, New York, NY, 334-342.
- [5] Carr, L., Miles-Board, T., Woukeu, A., Wills, G., and Hall, W. 2004. The case for explicit knowledge in documents. In Proceedings of the 2004 ACM Symposium on Document Engineering (Milwaukee, Wisconsin, USA, October 28 - 30, 2004). DocEng '04. ACM, New York, NY, 90-98.
- [6] Donaldson, J. J., Conover, M., Markines, B., Roinestad, H., and Menczer, F. 2008. Visualizing social links in exploratory search. In Proceedings of the Nineteenth ACM Conference on Hypertext and Hypermedia (Pittsburgh, PA, USA, June 19 - 21, 2008). HT '08. ACM, New York, NY, 213-218.
- [7] Grønbæk, K., Hem, J. A., Madsen, O. L., and Sloth, L. 1993. Designing Dexter-based cooperative hypermedia systems. In Proceedings of the Fifth ACM Conference on Hypertext (Seattle, Washington, United States, November 14 - 18, 1993). HYPERTEXT '93. ACM, New York, NY, 25-38.
- [8] Haake, J. M. and Wilson, B. 1992. Supporting collaborative writing of hyperdocuments in SEPIA. In Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work (Toronto, Ontario, Canada, November 01 - 04, 1992). CSCW '92. ACM, New York, NY, 138-146.
- [9] Hansen, F. A. and Grønbæk, K. 2008. Social web applications in the city: a lightweight infrastructure for urban computing. In Proceedings of the Nineteenth ACM Conference on Hypertext and Hypermedia (Pittsburgh, PA, USA, June 19 - 21, 2008). HT '08. ACM, New York, NY, 175-180.
- [10] Hsieh, H. and Shipman, F. M. 2002. Manipulating structured information in a visual workspace. In Proceedings of the 15th Annual ACM Symposium on User interface Software and Technology (Paris, France, October 27 - 30, 2002). UIST '02. ACM, New York, NY, 217-226.
- [11] Marshall, C. C. and Shipman, F. M. 1993. Searching for the missing link: discovering implicit structure in spatial hypertext. In Proceedings of the Fifth ACM Conference on Hypertext (Seattle, Washington, United States, November 14 - 18, 1993). HYPERTEXT '93. ACM, New York, NY, 217-230.
- [12] Nielsen, J. 2005. Reviving Advanced Hypertext. Jakob Nielsen's Alertbox, January 3, 2005, <http://www.useit.com/alertbox/20050103.html>
- [13] Rubart, J., Haake, J. M., Tietze, D. A., and Wang, W. 2001. Organizing shared enterprise workspaces using component-based cooperative hypermedia. In Proceedings of the 12th ACM Conference on Hypertext and Hypermedia (Århus, none, Denmark, August 14 - 18, 2001). HYPERTEXT '01. ACM, New York, NY, 73-82.
- [14] Shipman, F.M., Marshall, C.C., and Moran, T.P. "Finding and Using Implicit Structure in Human-Organized Spatial Layouts of Information." Proceedings of ACM CHI '95, Denver, Colorado (May 7-11, 1995), pp. 346-353.
- [15] Shipman, F. M., Hsieh, H., Maloor, P., and Moore, J. M. 2001. The visual knowledge builder: a second generation spatial hypertext. In Proceedings of the 12th ACM Conference on Hypertext and Hypermedia (Århus, none, Denmark, August 14 - 18, 2001). HYPERTEXT '01. ACM, New York, NY, 113-122.
- [16] Shipman, F. M. and Marshall, C. C. 1999. Spatial hypertext: an alternative to navigational and semantic links. ACM Comput. Surv. 31, 4es (Dec. 1999), 14.
- [17] Styliaras, G. D. and Christodoulou, S. P. 2009. HyperSea: towards a spatial hypertext environment for web 2.0 content. In Proceedings of the 20th ACM Conference on Hypertext and Hypermedia (Torino, Italy, June 29 - July 01, 2009). HT '09. ACM, New York, NY, 35-44.
- [18] Wang, W. and Haake, J. M. 2002. Supporting distributed meetings using cooperative, visual, process-enabled hypermedia. In *Proceedings of the Thirteenth ACM Conference on Hypertext and Hypermedia* (College Park, Maryland, USA, June 11 - 15, 2002). J. Blustein, Ed. HYPERTEXT '02. ACM, New York, NY, 147-148.
- [19] Yamamoto, Y., Nakakoji, K., and Aoki, A. 2002. Spatial Hypertext for linear-information authoring: Interaction design and system development based on the ART Design principle. In Proceedings of the Thirteenth ACM Conference on Hypertext and Hypermedia (College Park, Maryland, USA, June 11 - 15, 2002). J. Blustein, Ed. HYPERTEXT '02. ACM, New York, NY, 35-44.