



Project no. FP6-028038

## Palette

Pedagogically sustained Adaptive LEarning Through the exploitation of Tacit and Explicit knowledge

Instrument: Integrated Project

Thematic Priority: Technology-enhanced learning

**D.MED.06 – Second operational version of CoPe\_it!**

Due date of deliverable: January 31, 2008  
 Actual submission date: March 06, 2008

Start date of project: February 01, 2006

Duration: 36 months

Organisation name of lead contractor for this deliverable: CTI

|   |        |    |
|---|--------|----|
| Project co-funded by the European Commission within the Sixth Framework Programme |        |    |
| Dissemination Level   |        |    |
| P   | Public | PU |

Keyword List: Collaboration Services, Argumentative Collaboration, CoPe\_it!  
 Responsible partner: CTI

| <i>MODIFICATION CONTROL</i> |                   |               |                                      |
|-----------------------------|-------------------|---------------|--------------------------------------|
| <i>Version</i>              | <i>Date</i>       | <i>Status</i> | <i>Modifications made by</i>         |
| 1.0                         | February 15, 2008 | Draft         | CTI (version sent to the evaluators) |
| 2.0                         | February 25, 2008 | Draft         | CTI (version sent to the SC)         |
| 3.0                         | February 29, 2008 | Final         | CTI (version sent to SCO and AFC)    |
|                             |                   |               |                                      |
|                             |                   |               |                                      |

#### Deliverable manager

§ Nikos Karacapilidis, CTI

#### List of Contributors

§ Nikos Karacapilidis, CTI  
 § Manolis Tzagarakis, CTI  
 § Dora Nousia, CTI  
 § Vassilis Kallistros, CTI  
 § Spyros Christodoulou, CTI  
 § Nikos Karousos, CTI  
 § George Gkotsis, CTI  
 § Christos Mettouris, CTI

#### List of Evaluators

§ Amaury Daele, UNIFR  
 § Stéphane Sire, EPFL

## Executive Summary

This document reports on the second operational version of CoPe\_it! (v.2.2, release: February 28, 2008). It should be read as a continuation of the deliverable D.MED.02 ("First operational version of the web-based tool supporting argumentative collaboration towards learning"), which has described the functionality of the first operational version of the tool (v.2.03, release: June 21, 2007), the basic concepts behind it, as well as the various views and projections available for the visualization of the argumentative process. More specifically, this document focuses on the new functionality added in the tool's second operational version, while it also indicates amendments done towards improving the functionality of its first operational version (such amendments took into account the work carried out in the context of WP1 and WP4). In addition, the document gives two illustrative examples of the tool's use and outlines future development plans.

## Table of contents

|       |  |    |
|-------|--|----|
| 1     | OVERVIEW .....   | 4  |
| 2     | ADDITIONS AND IMPROVEMENTS .....                           | 5  |
| 2.1   | USER & ROLE MANAGEMENT ISSUES .....                        | 5  |
| 2.1.1 | Access Control .....                                       | 5  |
| 2.1.2 | Community creation .....                                   | 9  |
| 2.2   | AWARENESS & COMMUNICATION ISSUES .....                     | 10 |
| 2.2.1 | Minimap .....  | 10 |
| 2.2.2 | Communication by e-mail .....                              | 11 |
| 2.2.3 | Tag Cloud view .....                                       | 11 |
| 2.3   | COLLABORATION WORKSPACES AND OBJECTS .....                 | 12 |
| 2.3.1 | New Content Types for Collaboration Objects .....          | 12 |
| 2.3.2 | Transformation rules .....                                 | 14 |
| 2.3.3 | Collaboration Workspace thumbnails .....                   | 15 |
| 2.3.4 | Collaboration Workspace properties .....                   | 16 |
| 2.3.5 | Object properties .....                                    | 16 |
| 2.4   | LEGACY RESOURCES & INTEROPERABILITY ISSUES .....           | 17 |
| 2.4.1 | Forum import in Collaboration Workspace .....              | 17 |
| 2.4.2 | eLogbook integration .....                                 | 19 |
| 2.4.3 | Compendium interoperability .....                          | 20 |
| 2.4.4 | Integration of an HTML Editor .....                        | 22 |
| 2.5   | REPOSITORY ADDRESSING MECHANISM .....                      | 22 |
| 2.6   | OPTIMIZING PERFORMANCE .....                               | 23 |
| 2.6.1 | Relation Drawing .....                                     | 23 |
| 3     | USE CASES .....  | 26 |
| 3.1   | USING CoPE_IT! IN DATA-INTENSIVE COLLABORATION .....       | 26 |
| 3.2   | USING CoPE_IT! IN A COMMUNITY OF EDUCATORS .....           | 30 |
| 4     | FUTURE DEVELOPMENT PLANS .....                             | 37 |
| 5     | REFERENCES .....   | 40 |
|       | APPENDIX A: SOME FIGURES ABOUT THE TOOL'S ACTIVE USE ..... | 41 |
|       | APPENDIX B: LICENSE & CODE INFORMATION (UPDATED) .....     | 42 |
|       | APPENDIX C: GOOGLE ANALYTICS .....                         | 44 |

# 1 Overview

CoPe\_it! (<http://copeit.cti.gr>) is a web-based tool supporting argumentative collaboration towards learning, which is being developed in the context of the Palette project. The tool implements a series of functional specifications of argumentative collaboration services, which have been described in detail in D.MED.05 ("Functional Specifications of Argumentative Collaboration Services"). It addresses the related CoPs' needs, as these have been identified and continuously elaborated through the work performed in the context of WP1 and WP5. Moreover, the tool is being developed in line with the Web services approach followed throughout the project; the technical specifications of its Web services have been presented in D.MED.09 ("Technical specifications of collaboration support tools as Web services").

The first operational version of CoPe\_it! (v2.03, release: June 21, 2007) was developed from scratch. In a later version (v2.1, release: October 19, 2007), we expanded the functionality offered and made progress on the tool's performance and user interface. In the recently released second operational version (v2.2, release: February 28, 2008), we have further improved the tool's functionality, usability and performance, by elaborating issues related to the user and role management, awareness and communication, collaboration workspaces and objects, exploitation of legacy resources, interoperability with other tools, and the tool's resource repository.

This document should be read as a continuation of the deliverable D.MED.02 ("First operational version of the web-based tool supporting argumentative collaboration towards learning"). It reports on the new functionality added in the tool's second operational version, while it also indicates amendments done towards improving the functionality of its first operational version. Following the participatory design methodology adopted in Palette, such amendments took into account the usability analysis described in D.PAR.04 ("User-centered description of PALETTE tools and services, and first analysis of usability"). More specifically, amendments done after the related suggestions of D.PAR.04 mainly concern the improvement of the tool's performance (e.g. when opening and working on workspaces), and various User Interface issues (e.g. consistency in the user interface elements used)\*.

In addition, this document gives two illustrative examples (use cases) of the tool's use (demonstrating the applicability of a series of CoPe\_it! functionalities) and outlines future development plans.

Developers involved in the development of CoPe\_it! are: Manolis Tzagarakis, Nikos Karousos, Spyros Christodoulou, George Gkotsis, Vassilis Kallistros, Christos Mettouris and Panayiotis Kyriakou.

---

\* It is noted that this usability analysis concerned version 2.03, released on 21/06/2007.

## 2 Additions and improvements

CoPe\_it! supports, among others, diverse formalization and visualization features, argumentation mechanisms, management of diverse collaboration objects (stored in the content repository), decision making, exploitation of legacy resources, interoperability with other tools (developed in or out of the context of the Palette project), and a sophisticated management of users, Communities of Practice (CoPs), as well as their roles and privileges.

This section reports on additions to and improvements on the tool's first operational version (v2.03), which now appear in the recently released second operational version (v2.2). Referring to the contents of this section, the following table indicates which feature/functionality has been either added or improved.

| FEATURE/FUNCTIONALITY                             | ADDED | IMPROVED |
|---|-------|----------|
| 2.1.1 Access Control                              | √     |          |
| 2.1.2 Community creation                          |       | √        |
| 2.2.1 Minimap                                     | √     |          |
| 2.2.2 Communication by email                      | √     |          |
| 2.2.3 Tag Cloud view                              | √     |          |
| 2.3.1 New Content Types for Collaboration Objects |       | √        |
| 2.3.2 Transformation rules                        |       | √        |
| 2.3.3 Collaboration Workspace thumbnails          | √     |          |
| 2.3.4 Collaboration Workspace properties          | √     |          |
| 2.3.5 Object properties                           | √     |          |
| 2.4.1 Forum import in Collaboration Workspace     | √     |          |
| 2.4.2 eLogbook integration                        | √     |          |
| 2.4.3 Compendium interoperability                 | √     |          |
| 2.4.4 HTML Editor integration                     | √     |          |
| 2.5 Repository addressing mechanism               | √     |          |
| 2.6.1 Relation Drawing (optimizing performance)   |       | √        |

Table 1: Features / functionalities added or improved

### 2.1 User & Role management issues

#### 2.1.1 Access Control

According to the project's internal document entitled "Preparation of the Task Force for the regulation and the reorganisation of the teams' work in WP5" (see document entitled "Proposal for the preparation

of the Task Force” at <https://bscw.ercim.org/bscw/bscw.cgi/334477>, last modified: Oct 8, 2007), the following two needs have been identified as very important for CoPs: the need to improve the exploitation of the common resources and the need to improve the participation and its quality. These needs are partially (explicitly or implicitly) associated to the ability to organise the activities with different possible levels of participation. Towards this direction, CoPe\_it! encompasses an Access Control module, which allows the members of CoPs to assume different roles in their interaction with the system.

### *Conceptual Model*

CoPe\_it! has implemented a full-fledged access-control system based on privileges. For easiness-of-use purposes, we have also implemented a set of predefined roles. Let us first start with some definitions:

- User: a CoPe\_it! system user
- Group: a CoPe\_it! system group; a Group has Users as members.
- Holder: (i.e. the Holder of a Privilege) maybe a User, a Group or a Role (see below for definitions of Privilege and Role).
- System Objects (SO): CoPe\_it! system entities. Each of them supports specific actions. System Objects are predefined in the system but may be augmented (if the need for new entities arises). System Objects, at present, are the following: System, CoP, Collaboration Workspace, Document, Link, Adornment (a visual cue serving the grouping of Documents), User, Group, Role.
- System Object instance (SOi): an instantiated System Object (i.e. the CoP “Earth” is a System Object instance of the CoP System Object type).
- Action: a distinct operation on a System Object. Actions refer to specific System Objects, are predefined but may be changed programmatically (i.e. CREATE\_COP [the creation of a new CoP] is an example of an Action on the CoP System Object).
- Privilege: is a triplet (Action, System Object instance, Holder) defining that a Holder is authorised for an Action and this authorisation is given on a specific SOi (a more detailed description of the impact that this has on the Access Control System is given in the following section). Note that Action may be any Action, not just those supported by the SO. A Privilege has also a number of attributes, such as SkipParent, User who granted the Privilege, Date of creation etc. (for details, see below).

## Design

Most Role-based Access Control Systems support Role Hierarchies [Ferraiolo *et al.*, 2001]. A Role Hierarchy is a partial order on roles called the inheritance relation, denoted as  $\geq$ , where  $r_1 \geq r_2$  only if all permissions of  $r_2$  are also permissions of  $r_1$ , and all users of  $r_1$  are also users of  $r_2$ . These systems usually support Limited Role Hierarchies in which a Role may inherit from only one other Role. Role Hierarchies are appropriate for systems which support an organization with multi-level authority and responsibility structure. In CoPe\_it!, we do not have such an organizational structure but, on the other hand, we have a hierarchical data structure (the System contains a number of CoPs, each of which contains a number of Collaboration Workspaces, each of which consists of Documents, Links and Adornments); moreover, we need flexibility and fast response times. The Access Control System of CoPe\_it! relies on System Object Hierarchy where a child node inherits all the privileges of its parent. So the scope of a Privilege located on a SOi is all the descendent instances of this SOi. The hierarchy of the SO is presented in Figure 1 (the System SO has only one SOi and is the common ancestor of all SOi).

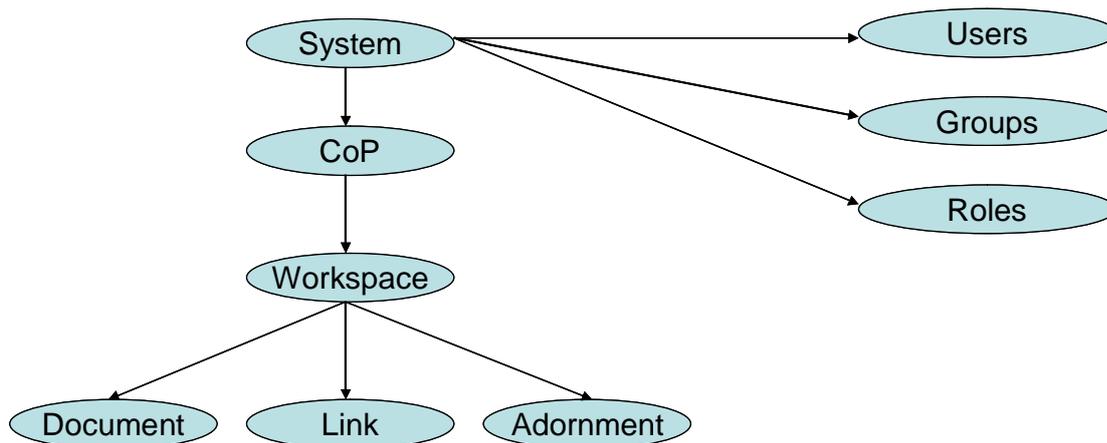


Figure 1: Hierarchy of CoPe\_it! System Objects

As an example, let us suppose that the Privilege: (UPDATE\_DOCUMENT, Earth CoP, User1) is granted. Let us first clarify that the action UPDATE\_DOCUMENT is an action which may be performed on a Document SOi. Consequently, User1 has the permission to Update all Documents located in all Collaboration Workspaces of CoP "Earth". On the other hand, the Privilege: (UPDATE\_DOCUMENT, Welcome workspace, User2) allows User2 to Update all documents located only in Collaboration Workspace "Welcome". The *SkipParent* attribute of a Privilege P denotes that Privileges with the same Action as that of P are not inherited from upper levels of the hierarchy.

At this point, we may introduce the notion of Role in CoPe\_it!. Role is a set of tuples (Action, SkipParent, Location). A Role is actually a template to be used in the creation of a collection of privileges. So, a Role defines Actions and for each one it specifies the *SkipParent* attribute of the produced Privilege, as well as the location in the Hierarchy on which the produced Privilege will be placed. Note that Role is itself a SO, so it supports Actions and has attached Privileges. Roles allow the convenient handling of predefined sets of privileges.

There are three stages in the deployment of a Role: definition, application, assignment.

- Definition: In this stage, the tuples contained in the Role are defined.
- Application (instantiation) on a SOi: The application of a Role produces the Privileges as specified in its definition and a Role Instance is created. The Holder of those Privileges is set as the Role instance itself.
- Assignment of a Role Instance to a User or Group: makes the user/group member of the Role Instance.

The history of assignment of Users to Role Instances is kept in the system, so it is possible to examine past roles which users have assumed.

We have currently defined the following Roles: *Visitor* to a CoP (with permission to only view Collaboration Workspaces), *CommunityMember* (with permission to add objects to Collaboration Workspaces but not modify existing ones), *workspaceModerator* (with all actions granted on a specific Collaboration Workspace), *CommunityLeader* (with all actions granted on a specific CoP), *SystemAdministrator*, *PrivateworkspaceViewer* (with permission to only view a *private* Collaboration Workspace), and *PrivateworkspaceOwner* (with all actions granted on a specific *private* Collaboration Workspace).

New Roles may be defined in the System and existing ones may be modified or deleted. In the current version, Roles are applied and assigned programmatically, but in future versions users will be able to handle them through the User Interface.

To recapitulate, when a user needs to perform an Action A on a SOi, the Access Control module checks for:

```
Existence of a Privilege on the Action A located on the SOi where the holder
is either the user, or the groups the user is member of, or the roles the
user is member of, or the roles the user-groups are members of.
If
    no such Privileges were found AND
```

there doesn't exist a Privilege for the Action A (with any holder) with the SkipParent attribute set

Then The search is propagated upwards in the system objects hierarchy

Else If such Privileges were found the user is granted permission otherwise not

*Summing up*

The basic unit of the CoPe\_it! Access Control module is the Privilege. Privileges are inherited in the System Object hierarchy unless the *SkipParent* attribute is set. Because of the inheritance in Privileges, a small number of Privileges need to be defined for handling a much larger number of SOi. Although it is possible to grant specific privileges to users, the most convenient way to do so is through Roles. Roles are templates for the "mass" creation of Privileges and are more amenable for handling by end users. In any case, new Roles may be defined, while existing ones may be modified or deleted (our approach is generic enough to accommodate any combination of privileges on any CoPe\_it! object).

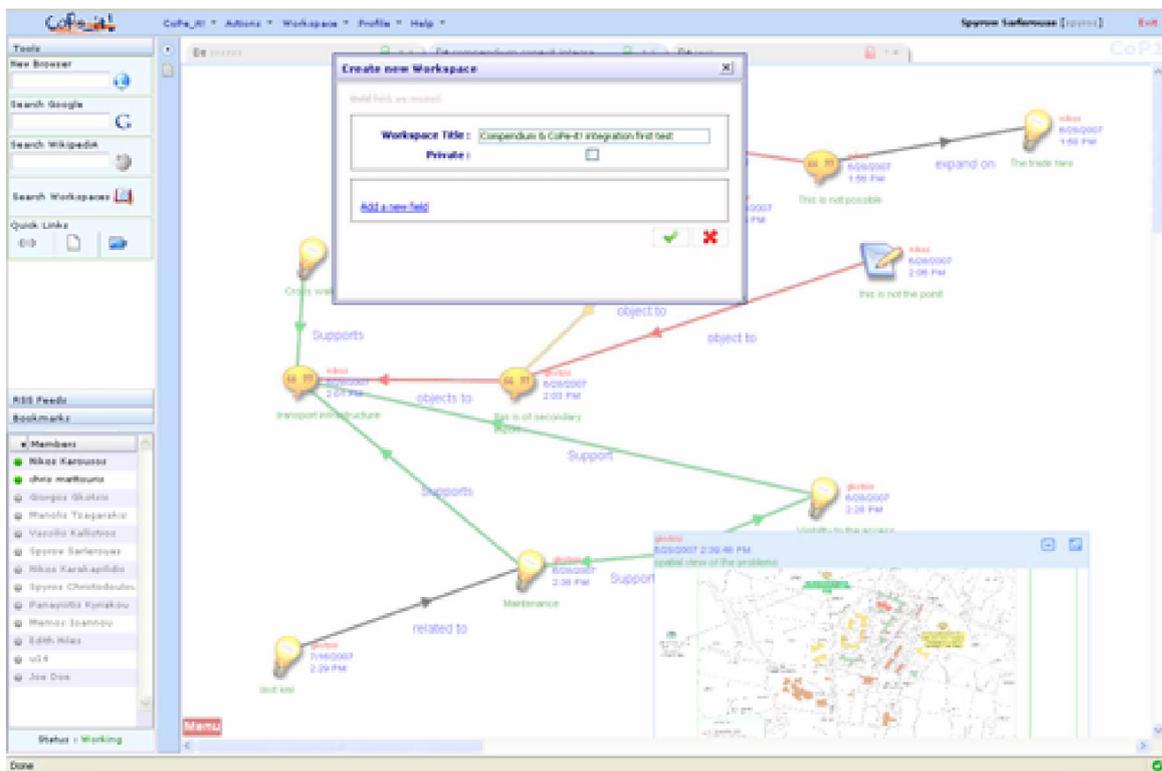


Figure 2: Community creation

2.1.2 Community creation

Community creation was a privilege granted only to specific CoPe\_it! users in the tool's first operational version. In the second operational version, every CoPe\_it! user is able to create his own Communities,

as long as he has read and agreed to a Disclaimer Note. Community creation can be performed by selecting CoPe\_it! → Create New Community option on the CoPe\_it! top menu (Figure 2) .

Community creation (and deletion) is also provided as a Web service. For further details, refer to D.MED.09 (“Technical specifications of collaboration support tools as web services”).

## 2.2 Awareness & communication issues

### 2.2.1 Minimap

Every Collaboration Workspace in CoPe\_it! is equipped with a mini map providing a full overview of it (Figure 3, bottom right part). Within the mini map, the user is able to see areas of activity of a Collaboration Workspace that indicate which issues are being discussed by other community members. It also enables users to see where the most collaborative activity is happening. While the main display permits a user to focus on his/her own tasks, the mini map lets the same user glimpse on which issues other members of the Collaboration Workspace are currently engaged. In different situations, this may also provide valuable insights that facilitate the coordination of collaborative actions within a Collaboration Workspace. Finally, it can be used for navigation purposes, to directly shift one’s view on another area of the workspace.

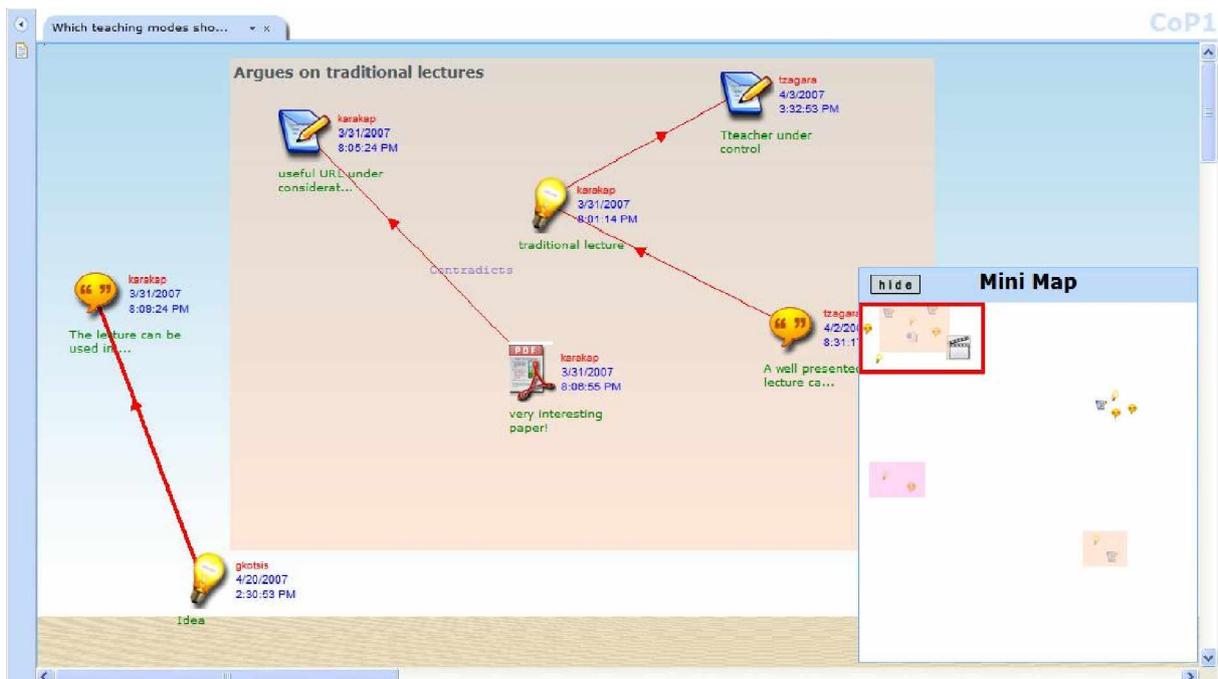


Figure 3: Mini map giving an overview of the entire Collaboration Workspace

## 2.2.2 Communication by e-mail

E-mail communication with CoP members is supported. A user may contact a member of his CoP by clicking on his name on the members' area (Figure 4, bottom left pane). The "Send Mail" form appears, where the user has to fill in the title and the body of the e-mail.

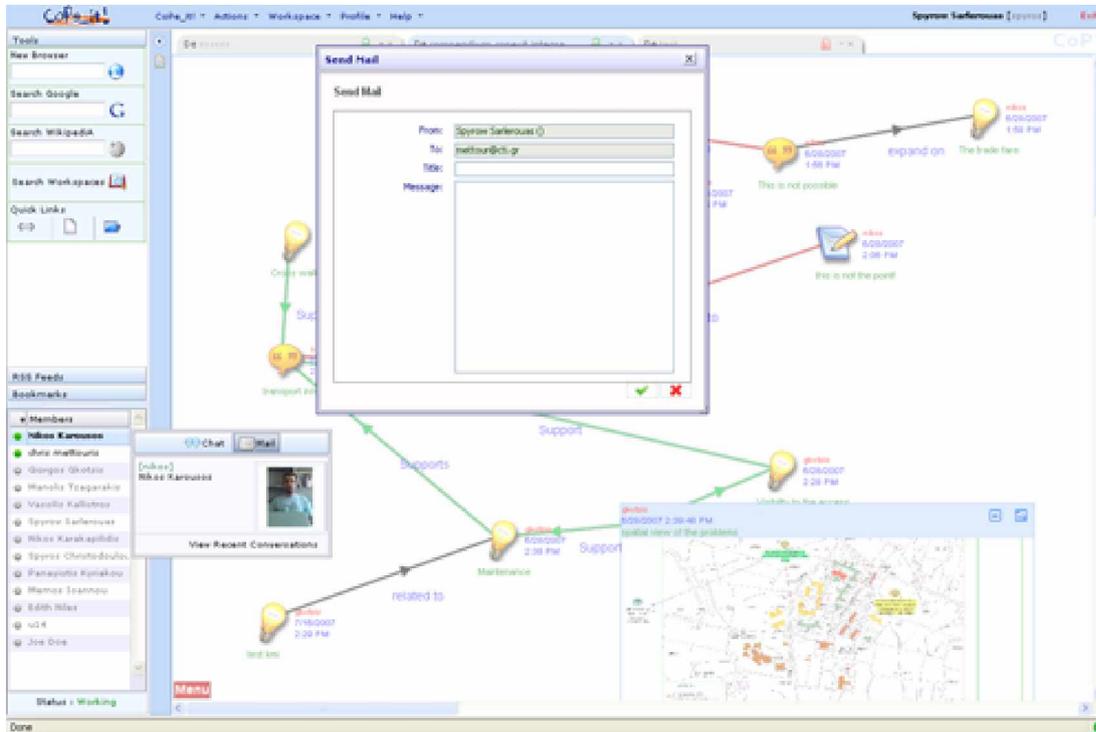


Figure 4: Communication by e-mail

## 2.2.3 Tag Cloud view

CoPe\_it! enables the visualization of a Collaboration Workspace in terms of the most frequently used terms, which constitute the tag cloud view. The tag cloud view uses visual cues to denote the significance of a term: more frequent terms appear in the tag cloud view with greater font size (Figure 5). The aim is to give a summary of the underlying collaboration and identify its current focus.

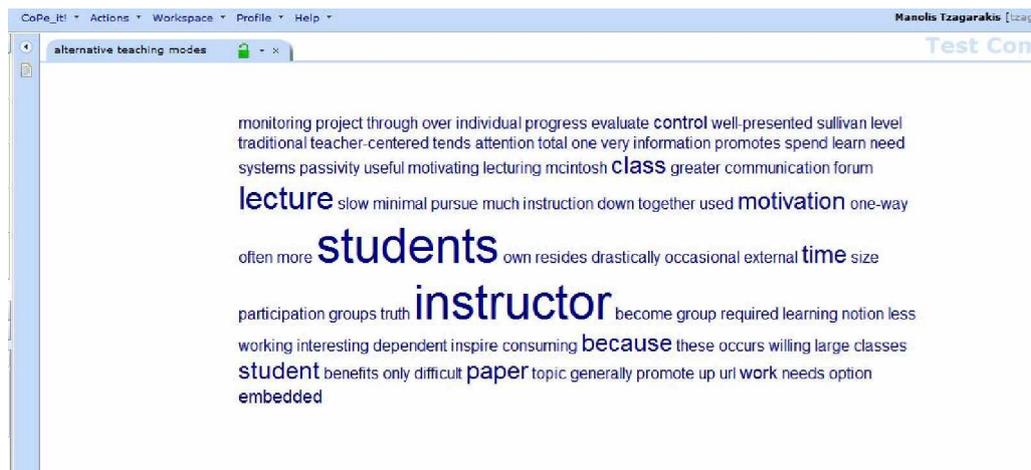


Figure 5: Tag cloud view of a Collaboration Workspace

In order to generate the tag cloud view, CoPe\_it! analyzes the text of each item on the Collaboration Workspace. The current implementation of the tag cloud view takes into consideration only the title of Collaboration Workspace items. Future versions of CoPe\_it! will also utilize the text body of each item as well as the content of existing files, and make this cloud navigational. The current algorithm does not take into consideration stop words. The list of such stop words can be customized to meet the CoP needs. The algorithm generating tag clouds does not depend on the language and hence a range of languages can be supported.

## 2.3 Collaboration Workspaces and Objects

### 2.3.1 New Content Types for Collaboration Objects

A Collaboration Object can be described by its knowledge type. In the first operational version of CoPe\_it!, a Collaboration Object's knowledge type could be: Document (file), Comment, Idea, Note, or Embedded object (such as an image or a video). By selecting the knowledge type of the Collaboration Object, the user was restricted to use a Collaboration Object content of a specific type (File, HTML text, image, video), i.e. when adding a Comment/Idea/Note Collaboration Object on the Collaboration Workspace, the user was forced to insert HTML text as content of the Collaboration Object. So, the knowledge type of the Collaboration object was restricting its content type.

Reviewing the functionality that the above knowledge types provided, we noticed that the user was not able to insert a File content or an image as a Note, an Idea or a Comment. This was certainly restrictive for the user. The user should be able to specify every Collaboration Object Content as a Note, an Idea or

a Comment. The answer to the above problem was to provide a new knowledge type for the Collaboration Object, the Unknown type. An unknown object can be anything that is not an Idea, a Note or a Comment. It may be of any data type and may be also used in cases the user is not certain about the knowledge type of his object or does not want to provide one. At the same time, we stopped using the terms Document and Embedded as knowledge types. Documents (or files of any kind) and Embedded objects (images/videos) fall in the category of Unknown type, if not specified otherwise, as all other Collaboration Objects.

The above arrangement can be seen more clearly by using a two-dimensional view of what a Collaboration Object might be (Figure 6 - the horizontal axis represents the available knowledge types and the vertical axis represents the actual type of content of a Collaboration Object).

| Knowledge Type \ Content Type | IDEA | NOTE | COMMENT | UNKNOWN |
|-------------------------------|------|------|---------|---------|
| TXT                           |      |      |         |         |
| HTML                          |      |      |         |         |
| IMAGE/VIDEO                   |      |      |         |         |
| JPG                           |      |      |         |         |
| XLS                           |      |      |         |         |
| DOC                           |      |      |         |         |
| URL                           |      |      |         |         |
| ...                           |      |      |         |         |

Figure 6: Available Collaboration Object Content and Knowledge Types

It is noted that the knowledge type of any object may be transformed into any other type (hence, the horizontal movement of a Collaboration Object in Figure 6 is feasible). To support the new knowledge type functionality, we properly modified the form used for adding new Collaboration Objects (the new form uses the two-dimensional view of Figure 6). As shown in Figure 7, the horizontal icons represent the available knowledge types, while the vertical icons represent the actual data type of an object, which may be Text, Url or File. Notice that an object of any data type may be set to be of any knowledge type.

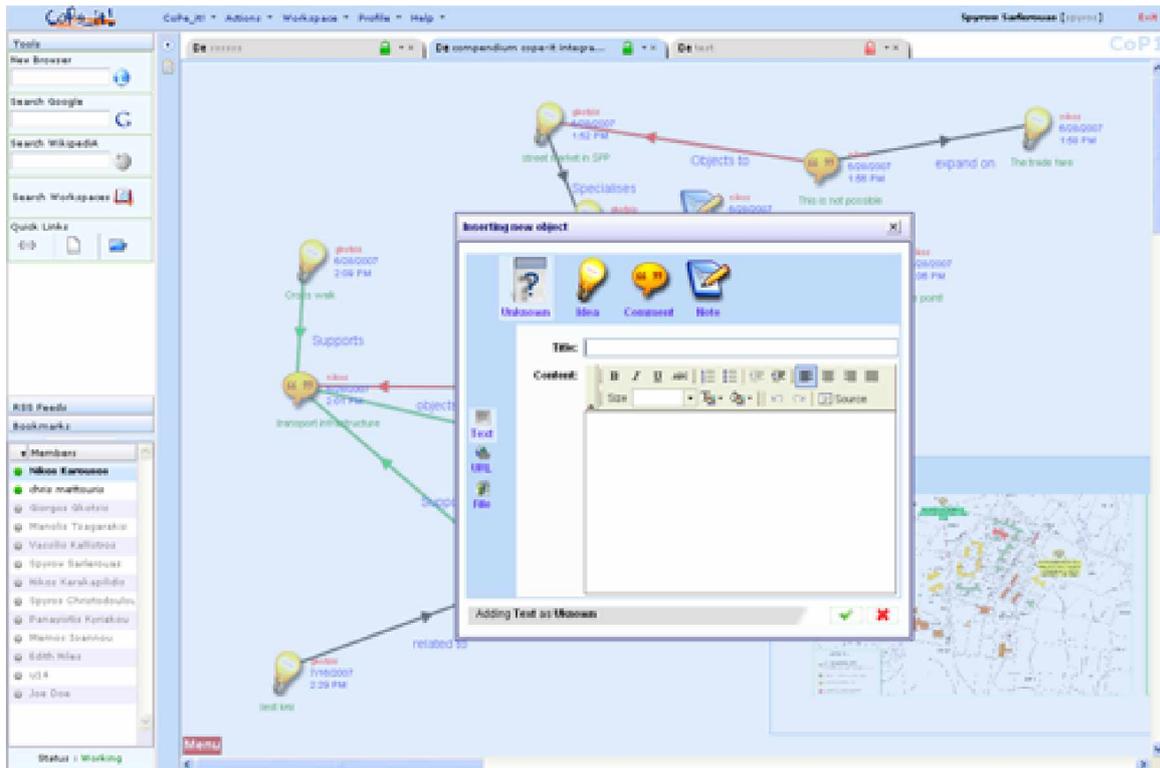


Figure 7: Adding a new Collaboration Object

The Text data type allows for adding and editing text in HTML format, by using the editor provided (see Section 2.4.4). Finally, the URL data type may be used for adding embedded images and videos, as well as links.

### 2.3.2 Transformation rules

During the Collaboration Workspace transformation from informal to formal, the user is prompted with an explanatory window describing the rules applied. New transformation rules now include a notation for neutral positions (i.e. arguments that are skipped and now shown neither as positive nor as negative arguments). More precisely, the current transformation rules are the following:

- Issue (root of the tree) is the name of the collaboration space
- Alternatives (nodes of 2nd level) are the ideas being produced in the desktop view
- All items linked to the ideas are parsed to complete the tree structure and are handled as positions. On the produced tree, the relationship parent-child may be either positive or negative (argument in favor or against). This information is extracted, through parsing relationships in the desktop view. This is performed by following some transformation rules:
  - Links with caption "support", "favor", "agree" are considered to be positive arguments.

- Links with caption "against", "defeats", "disagree" are considered to be negative arguments.
- Links with green color refer to a positive argument, whereas links with red color refer to a negative argument.
- Links with any other color are considered as "neutral" positions and are ignored.

The above reasoning mechanism is also provided as a Web service as it is not necessarily required to operate within CoPe\_it!. An application that exploits this service may upload an XML formatted document and the reasoning algorithm will produce the desired annotated document. For further details, refer to D.MED.09 ("Technical specifications of collaboration support tools as web services").

### 2.3.3 Collaboration Workspace thumbnails

In order to help participants to find (and open) a particular Collaboration Workspace as quickly as possible, CoPe\_it! provides a screenshot-based view of all available Collaboration Workspaces (Figure 8) by displaying a thumbnail image of the last saved state of each of them. To do so, CoPe\_it! captures pictures of the newly created or updated Collaboration Workspaces every 24 hours and converts them to thumbnails.



Figure 8: A thumbnail-based list of available Collaboration Workspaces

The thumbnail-view of Collaboration Workspace is available on the "Open workspace" option. It is worth mentioning that the above functionality is provided through the help of two external tools: ThumbnailList generator and URL2JPG (further details in Appendix B).

### 2.3.4 Collaboration Workspace properties

Users may view the properties of a Collaboration Workspace (Figure 9, left part) by selecting the appropriate action on the Collaboration Workspace menu (located on the tab of the Workspace).

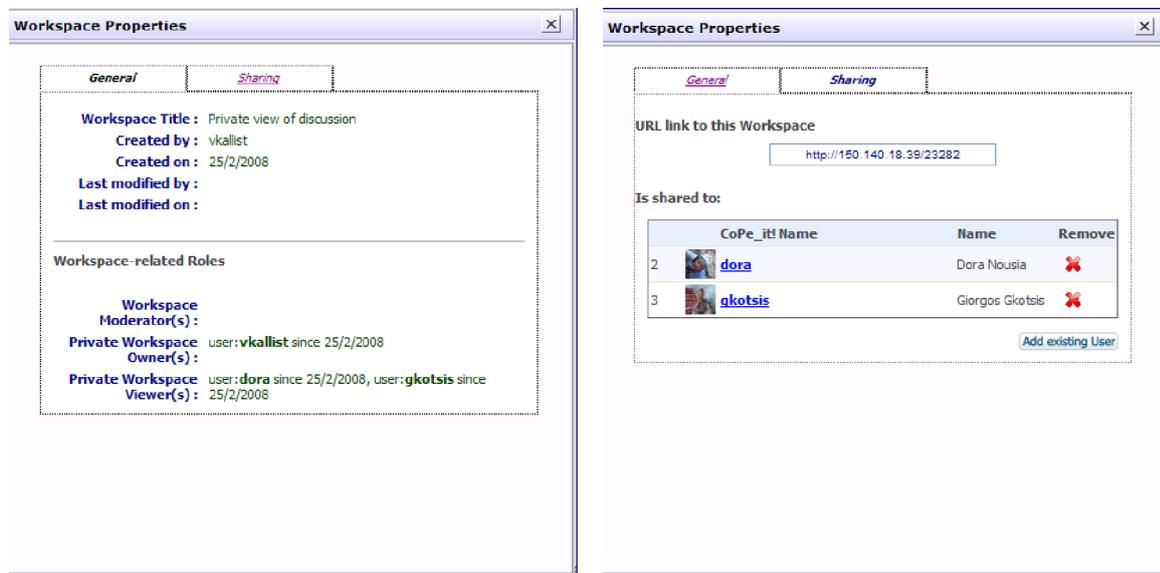


Figure 9: Collaboration Workspace properties

The properties window shows:

- The full title of the Collaboration Workspace;
- The user who created it and its creation date;
- The user who last modified an object inside it and when it was modified;
- The Workspace-related Roles: users/groups who have the Moderator, the Private Workspace Owner, or the Private Workspace Viewer roles. Users/groups who were members of the above-mentioned roles in the past are also displayed in grey.

Through a separate tab (Figure 9, right part), the owner of a private workspace may also allow its sharing (read only) with other CoPe\_it! members (these members will be notified through email about this action).

### 2.3.5 Object properties

Every Collaboration Object in the Collaboration Space has some Properties and Tags (Figure 10). Properties, such as the title and the creator of the object, its creation date, and its name (see D.MED.09), aim to inform the user about the object's features and characteristics. Each relation has also

properties including its creator, the date of creation, its width (width of the associated line) and the relation type (in favor/against/neutral). Tags, on the other hand, are used to describe the object and may provide hints about the meaning of its content and the purpose of its creation. Object and relation properties are fixed in the sense that they are not subject to any changes by the users. On the contrary, tags are user-defined and can be easily created and deleted.



Figure 10: Object properties and tags

## 2.4 Legacy resources & interoperability issues

### 2.4.1 Forum import in Collaboration Workspace

Towards providing capabilities to exploit legacy resources, CoPe\_it! enables users to import external discussion forums available on the World Wide Web into a Collaboration Workspace. More specifically, CoPe\_it! permits the import of topics found on discussion forums. The current version supports import of phpBB and Moodle based forums (this can be easily extended to also include other forum types). The decision to initially support the abovementioned forums was based on their popularity as well as their adoption by various Palette CoPs (e.g. ePrep forum at: <http://palette.ercim.org/forum/>).

When a user wishes to import a topic available on an external forum, he chooses the appropriate option from the main menu, and enters the target topic's URL along with the name of the Collaboration Workspace into which the discussion topic will be imported (Figure 11). The current version supports this import only into new Collaboration Workspaces (future releases will permit the import in existing ones).

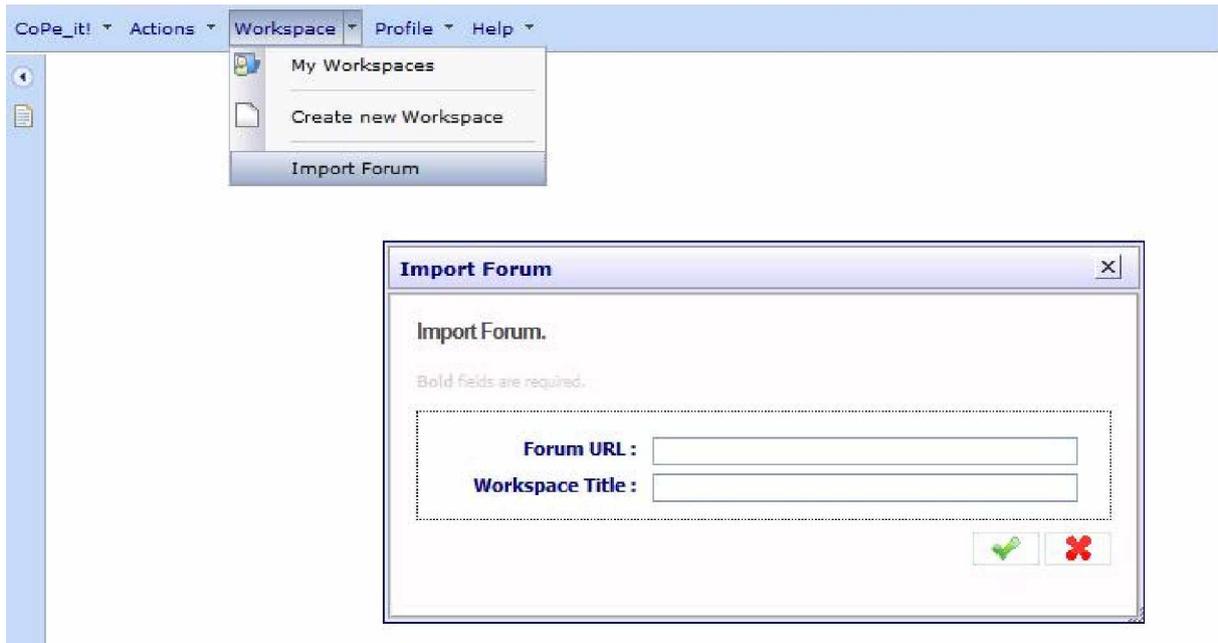


Figure 11: Import forum dialog in CoPe\_it!

Upon entering the target URL and executing the action, CoPe\_it! fetches all pages of the particular topic by essentially “crawling” the topic. Every retrieved topic page is parsed and every individual post is identified. All posts in the topic will appear in the new Collaboration Workspace as items of type “note” (Figure 12). All items that result from a forum import can be further processed like any other CoPe\_it! resource.

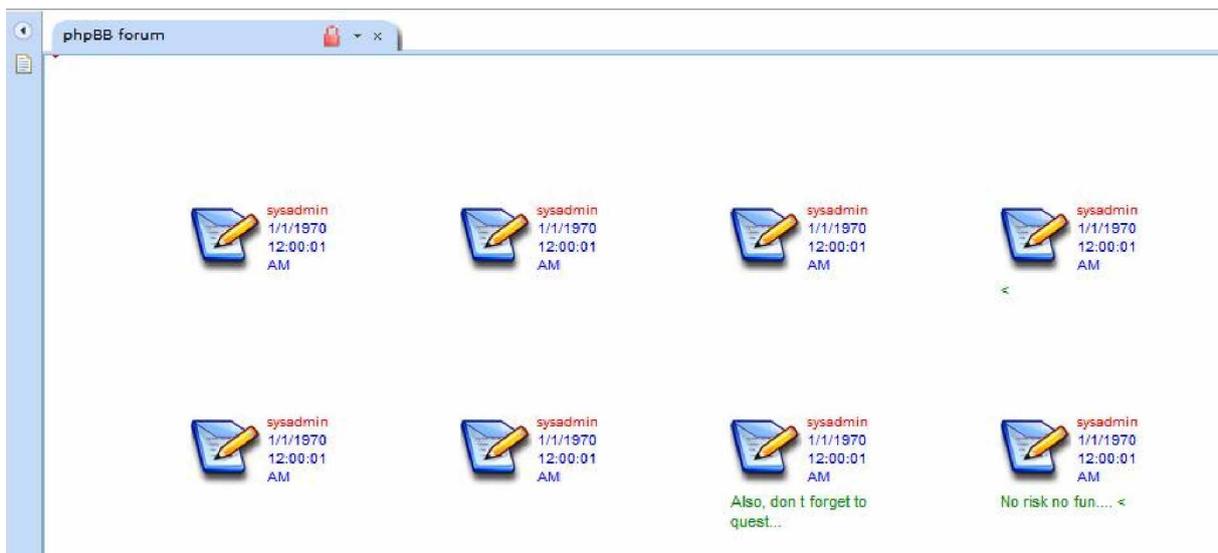


Figure 12: Items in a Collaboration Workspace as the result of invoking the import forum function; every individual item was a distinct post in the original topic (whenever applicable, the first characters of each post are displayed as the item’s title).

With respect to extensibility issues of the forums supported, the import functionality is based on a number of configuration settings that describe the structure of a forum page in terms of Regular Expressions (regex). Hence, support for additional forums can be achieved by simply providing regex expressions that capture all the necessary information of the target forum. The addition of such configuration files is not possible for end users.

## 2.4.2 eLogbook integration

One of the Collaboration Workspace views available to end users is the Context-Aware view, provided by eLogbook. This view is based on the Actor, Asset and Activity (AAA) model (described in D.MED.03: "First version of the eLogbook tool supporting exchange of artifacts at both the community and the technical level"). To provide this view, CoPe\_it! requests the appropriate services from eLogbook via RESTful communication.

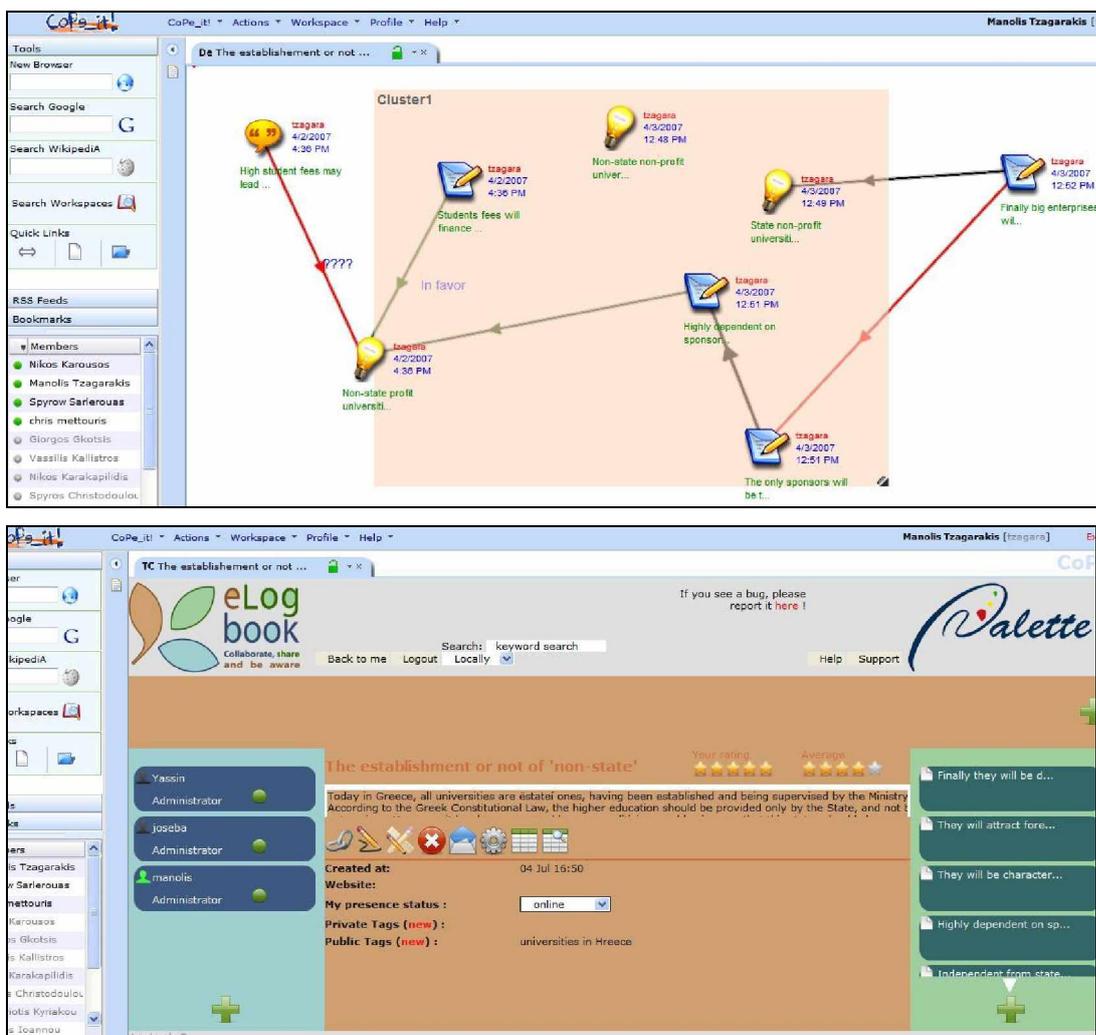


Figure 13: eLogbook integration in CoPe\_it! (top: the informal view of a particular Workspace; bottom: the Context-Aware view of the same Workspace)

In particular, an XML document representing the current Collaboration Workspace is sent to eLogbook which transforms the concepts found in the XML into eLogbook's AAA model. eLogbook renders the translated concepts in HTML and sends the results (a URL) back to CoPe\_it!, which in turn displays the result to the user (Figure 13). While navigation in the Context-Aware view is possible, no modification (addition, removal or update) can be done. More details about various CoPe\_it! – eLogbook integration issues can be found in D.MED.04: "First version of the awareness-support tool".

### 2.4.3 Compendium interoperability

One of the first test experiments towards CoPe\_it! evaluation as a collaboration tool was its comparison against popular products with similar functionality and goals. One of them is Compendium (<http://compendium.open.ac.uk/>), which has been widely adopted as a dialog mapping and meeting facilitation software product. Compendium was designed as a medium to accommodate discussions taking place in the real world (i.e. a group of people discussing face to face) and its main goal is the capturing of meeting discussions or collective definitions.

A thorough investigation and comparison between Compendium and CoPe\_it! led us conclude that it is useful to design and implement an automated transformation mechanism that enables the import and export of data from one tool to the other. In order to establish this interoperation, we studied the collaboration objects supported by each tool and sketched the related mapping, which is given below.

| CoPe_it!                              | Compendium |
|---------------------------------------|------------|
| Workspace                             | Map/View   |
| Idea                                  | Answer     |
| Comment                               | Argument   |
| Note                                  | Note       |
| Unknown / embedded / file (all types) | Reference  |

Based on a real collaboration session (the topic of the debate is "the contribution that the agriculture university can give to the Torre Guaceto community"), Figures 14 and 15 illustrate the particular Compendium "map" and its corresponding CoPe\_it! Collaboration Workspace, respectively (Figure 15 illustrates the Collaboration Workspace derived after the import of the Compendium "map" of Figure 14; export of a CoPe\_it! Collaboration Workspace and transformation of it into a Compendium "map" is also supported).



Figure 14: Compendium map

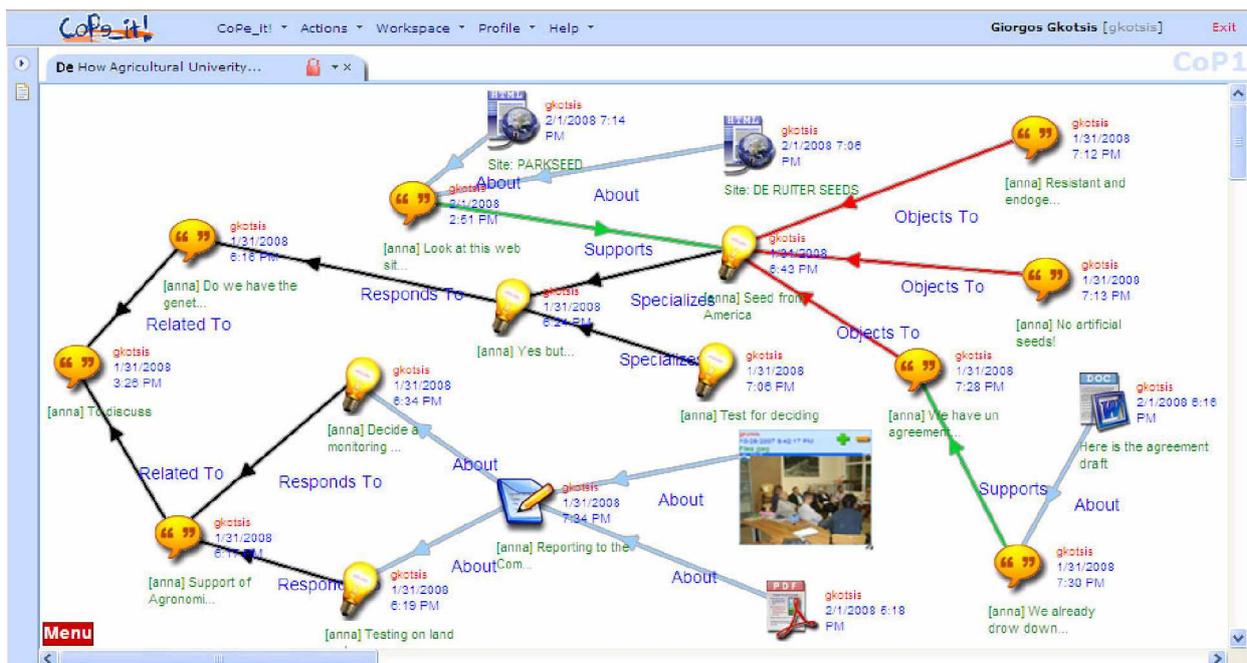


Figure 15: CoPe\_it! Collaboration Workspace

It is noted here that CoPe\_it! also provides an import/export mechanism as Web Service (refer to D.MED.09).

## 2.4.4 Integration of an HTML Editor

Having enabled a user to add a Collaboration Object of Content Type “HTML”, the associated requirement of integrating an appropriate HTML editor in CoPe\_it! came up. This would help users, especially the non expert ones, to create HTML formatted text easily and effectively. Among several HTML editors, the FCK Editor (Figure 16) was selected due to its popularity and easiness of use. For our purposes, only a subset of the FCK Editor features was used, mainly focusing on the potential needs of a CoPe\_it! user. These features are:

- Text Editing (Bold/Italic/Underline/Strikethrough/Size/Color/Background Color)
- Insert-Remove Numbered/Bulleted List
- Increase/Decrease indent
- Text Alignment (Left/Center/Right/Justify)
- Undo/Redo Typing
- Toggle HTML/HTML Source

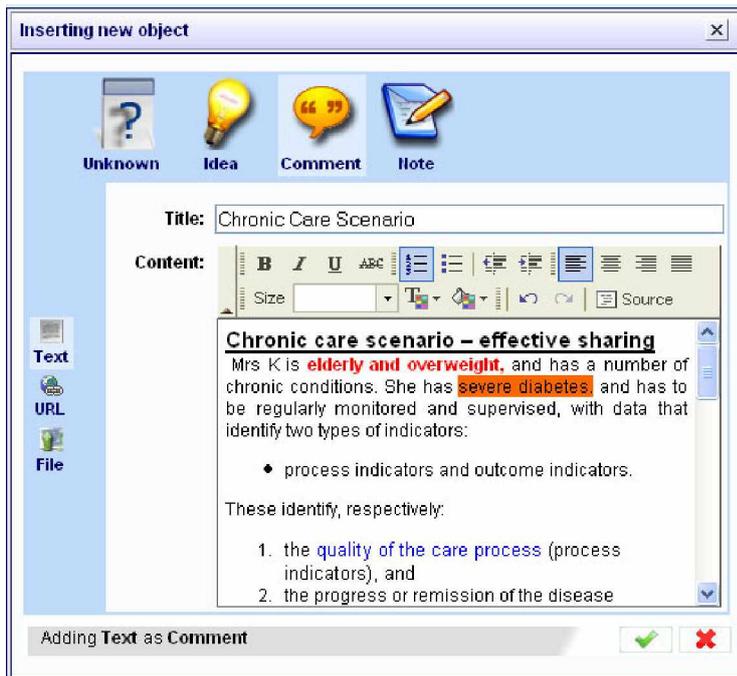


Figure 16: The HTML Editor in CoPe\_it!

## 2.5 Repository addressing mechanism

As sketched in D.MED.02, CoPe\_it! provides an object repository for the persistent storage and retrieval of objects (CoPe\_it! objects include resources uploaded on the tool's Collaboration

Workspaces and users). Some functions of CoPe\_it!'s repository will be available as independent or standalone services to external applications (see D.MED.09). This means that applications will be able to use these services independently of whether they make use of CoPe\_it!'s collaboration services or not. In other words, the Repository Services can be exploited separately from the tool's collaboration services. Nevertheless, their existence can increase the usefulness of the collaboration related services offered by CoPe\_it!.

One of the core features of CoPe\_it!'s repository is the addressing mechanism with which any object in the repository can be made addressable from any place on the Internet. The aim of the addressing mechanism is to permit easy referencing and sharing of CoPe\_it! resources by other tools. Towards this, CoPe\_it! builds on the URL concept and provides an addressing scheme that focuses on providing easy-to-remember, human readable URL strings. The *name* of a resource, a property available for all object types that is globally unique, but nevertheless specified by users during the creation of the resource, can be part of a CoPe\_it! resource address. For example, a workspace with the *name* "alternative teaching modes" can be referenced by the URL [http://copeit.cti.gr/alternative\\_teaching\\_modes](http://copeit.cti.gr/alternative_teaching_modes). Other resources will be able to be referenced by similar URLs.

Retrieval of objects, given a reference, is constrained by the tool's RBAC (Role Based Access Control) model. This means that when using a reference to retrieve (view) a resource, access will be granted only in case that the user has the appropriate privileges. Access to resources marked as public will be accessible to anyone, even to users and applications that have not been authenticated in CoPe\_it! (i.e. are considered as 'guests').

## 2.6 Optimizing performance

### 2.6.1 Relation Drawing

As noted in D.MED.02 ("First operational version of the web-based tool supporting argumentative collaboration towards learning"), several tests have been performed in order to optimize the load time of the tool's Collaboration Workspaces. Experimentations have indicated that Collaboration Workspaces including a large number of relations need more time to load than Collaboration Workspaces with a small, or even no, number of relations.

In the first operational version of CoPe\_it!, the method used for drawing relations (lines, from the browser's perspective) actually used pixel level drawing. In such methods, lines are decomposed in the pixels they consist of and the browser draws all the line pixels to form the requested line. Pixel level drawing methods have a number of advantages, among which the most important are that they are simple and short methods, allowing cross browser support. On the other hand, the main disadvantage of such methods is that they are slow and generate a lot of useless (HTML) tags. In our case, this results in making relation drawing/moving/editing and Collaboration Workspace loading slow. The need for a more effective drawing method was more than needed.

Several other methods have been tested in order to find the most appropriate one for CoPe\_it!. Figure 17 illustrates a comparison of applying the most common methods (pixel drawing, image loading, SVG and Canvas) used for line drawing to the CoPe\_it! case.

|               | Browser support |    |       | Browser Drawing Speed |      |       |
|---------------|-----------------|----|-------|-----------------------|------|-------|
|               | IE              | FF | Opera | IE                    | FF   | Opera |
| Pixel drawing | v               | v  | v     | Slow                  | Slow | Slow  |
| Image Loading | v               | v  | v     | Slow                  | Slow | Slow  |
| SVG           | -               | v  | v     | Fast                  | Fast | Fast  |
| Canvas        | v               | v  | v     | Fast                  | Fast | Fast  |

Figure 17: Drawing methods comparison in different browsers

Although pixel drawing methods ensure cross browser compatibility, they were slow in all three browsers tested. Image loading methods, based on loading an image to draw a line, were not only slow but also not applicable in CoPe\_it! (a few thousands of images would be required due to the fact that each CoPe\_it! relation may have one out of eight colors, one out of three widths and one out of 360 angles, not to mention that relation length is user defined). SVG (Simple Vector Graphics) is pretty fast but not supported in IE (plug-in required). The best method for CoPe\_it! was proved to be Canvas Tag method; this method is quite fast, natively supported by browsers such as Firefox/Opera and also supported in IE (using an open source library - Google's ExplorerCanvas project, based on IE's GVL support for shape drawing).



Figure 18: Collaboration Workspace loading time with variable relations number (Collaboration Workspace items = 30).

By using the Canvas Tag method for relation drawing in CoPe\_it!, actions such as relation drawing/editing and Collaboration Workspace loading became much faster than before. Experiments turned out that the average time needed to draw a relation is about 0.35 sec, while the corresponding time needed to draw a relation in CoPe\_it! first operational version was about 1.5 sec (see also Figure 18)<sup>†</sup>.

Applying the Canvas Tag method for drawing lines to CoPe\_it! had significant effects not only on Collaboration Workspace loading but also on the user friendliness of the environment. In the first operational version of CoPe\_it! , when creating a relation, the user had to select both the start and the end Collaboration object of the Collaboration Workspace to see the relation being formed. While dragging a Collaboration object on the Collaboration Workspace, its adjacent relations (if any) remained still to the point where dragging started and were moved, at once, to the point where object dragging ended. In the second operational version, due to the much less time and effort (from the browser's perspective) needed to draw a relation, the actions of creating a relation and dragging a Collaboration Object are more real-time (provide more graphical feedback). When creating a relation, a temporarily relation is drawn starting from the first Collaboration Object and finishing where the mouse points, resulting in making it (for the user) more obvious that a second Collaboration Object has to be selected (clicked) to create the permanent relation. When dragging an Object, its adjacent relations are dragged following the Object while dragged.

<sup>†</sup> Experiments were carried out on a Pentium IV, 3.2 GHz CPU, 1 GB RAM machine (Windows XP).

## 3 Use cases

### 3.1 Using CoPe\_it! in data-intensive collaboration

This section demonstrates the applicability of CoPe\_it! in a real, data-intensive debate that took place at the 5663<sup>rd</sup> Meeting of the United Nations Security Council (UNSC). An accurate representation of this debate has been incrementally built in CoPe\_it! by using its minutes (the official press release is available at: <http://www.un.org/News/Press/docs/2007/sc9000.doc.htm>). Additional data sources, corresponding to previous debates and information that is strongly related to the issue at hand, have been uploaded and appropriately linked in the corresponding workspace.

The context of this debate is as follows: United Kingdom, holding the presidency of the UNSC for April 2007, raised the issue of whether climate change has an impact on peace and security. Over fifty delegates from the UN Member States collaborated by expressing their points of view through various statements and arguments of legal, environmental, scientific and political content. Some delegates welcomed the initiative, while others expressed their doubts on the mandate of the UNSC to discuss such issues. Some arguments were in favor or against an idea or a previously asserted argument. In some cases, delegates were speaking on behalf of more than one member states (cases of geographically, politically or economically related states). Last but not least, delegates often referred to diverse sources of massive and complicated data such as data on territorial changes and maps, meteorological data, facts and figures related to energy resources, data concerning international agreements and protocols, financial data of different nature and complexity, records of similar debates conducted inside and outside UNSC, etc.

In the following, we sketch an emulation of the above debate through CoPe\_it! to show the tool's capabilities in tackling such data-intensive collaboration settings. The particular emulation was conducted in an asynchronous way, among geographically dispersed participants (taking the roles and using the wording of the real delegates). The layout of the tool's informal projection is shown in Figure 19. The left hand side bar enables participants to open a new browser, search for related information, and be aware of other online members of their community. Participants may easily create and upload various types of data and knowledge items (a predefined set of item types is given; participants may enrich this set by defining additional ones). These can be dedicated item types such as ideas (depicted as light bulbs), notes and comments, or existing multimedia resources. Ideas stand for items that deserve

further exploitation; they may correspond to an alternative solution to the issue under consideration and they usually trigger the evolution of the collaboration. Notes are generally considered as items expressing one's knowledge about the overall issue, an already asserted idea or note. Finally, comments are items that usually express less strong statements; they are uploaded to express some explanatory text or point to some potentially useful information. Knowledge item types may change upon the evolution of the collaboration (e.g. a user that has asserted a particular comment may elaborate it further and change its type to an idea). All the above items can be interrelated. When interrelating items, participants may select the color of the connecting arrow and provide a legend describing the interrelationship they conceive. CoPe\_it! enables participants to spatially arrange the uploaded items and organize them in a meaningful way.

Figure 19 corresponds to an early instance of the collaborative workspace created for the needs of our example. As shown, some stakeholders have contributed so far by uploading on the workspace some useful resources (including a video), proposing two concrete ideas, and interrelating one idea with four additional items.

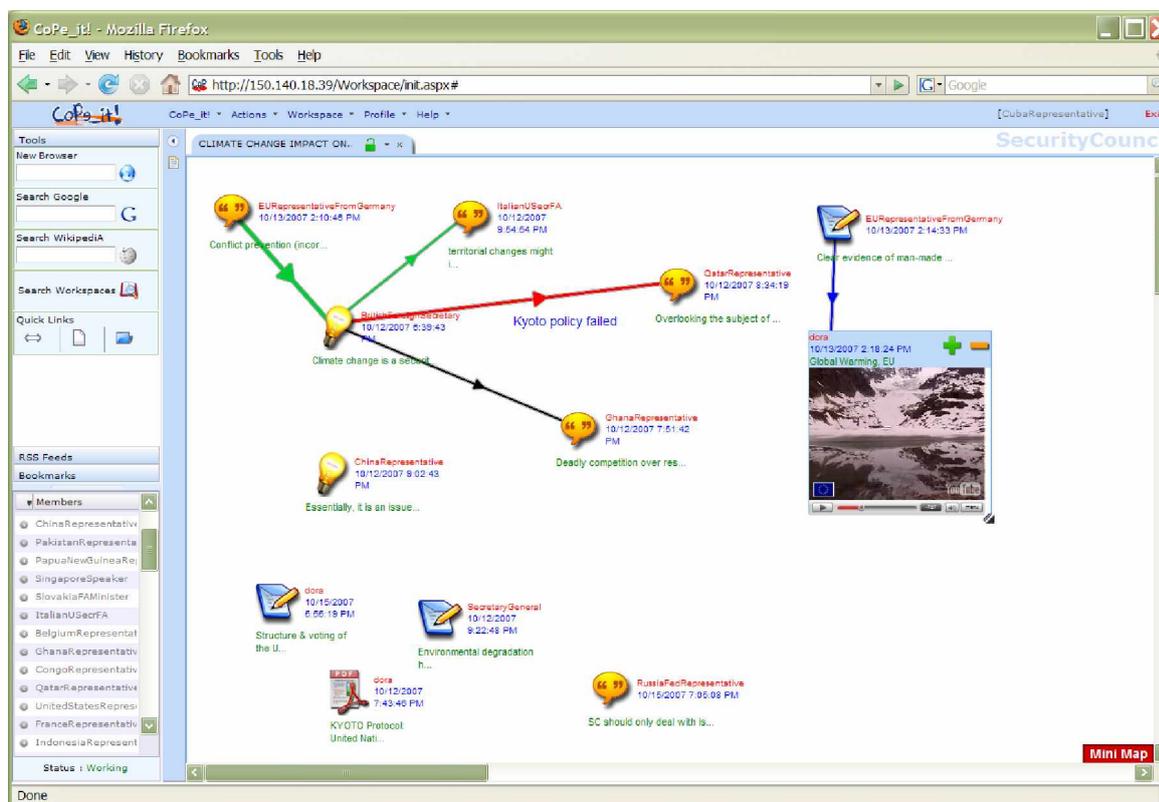


Figure 19: An early instance of the collaboration.

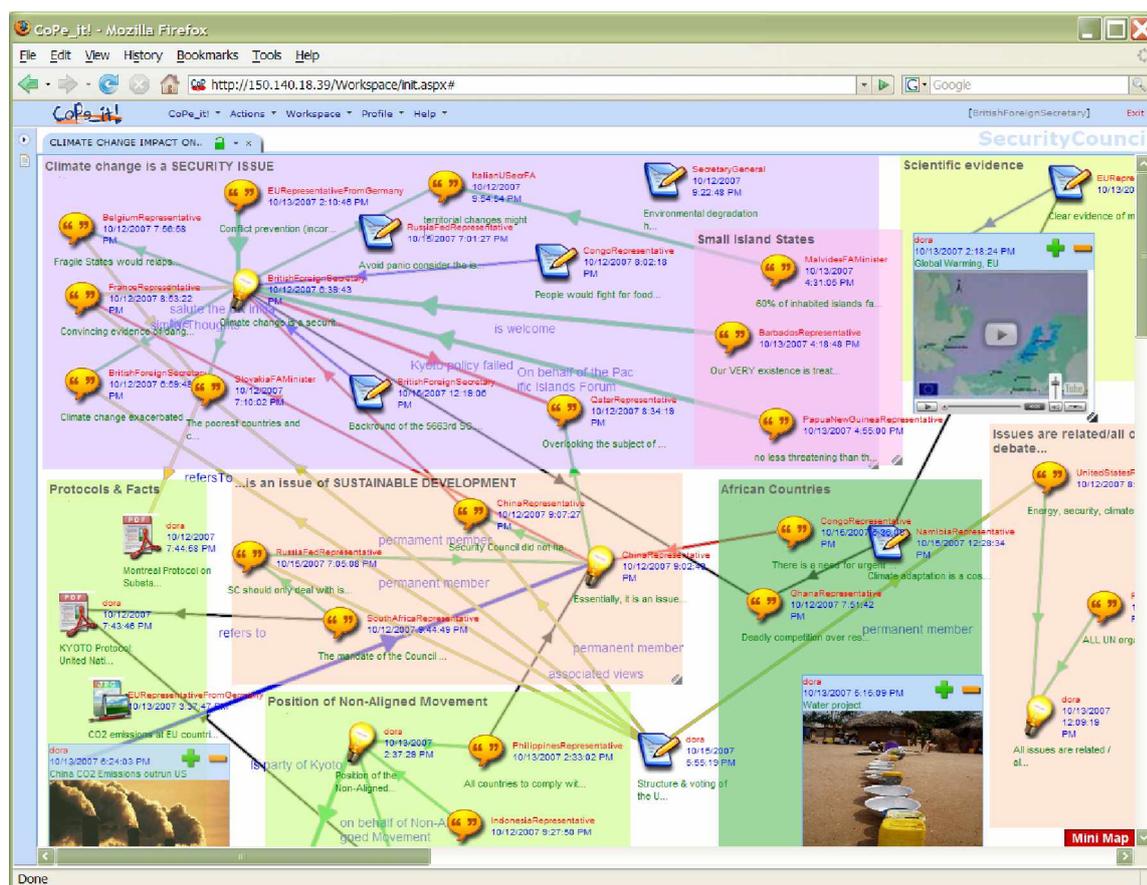


Figure 20: A data-intensive instance of the collaboration.

Figure 20 illustrates a second instance of the collaborative workspace under consideration (the left-hand side bar is closed for visualization purposes). As shown, this is a highly data-intensive instance where many delegates have been contributed their ideas and positions. Four ideas are now expressed, which are highly interconnected with multiple data and knowledge artifacts. More multimedia resources, particularly relevant to some items, have been also uploaded. Beyond coloring of the arrows that interrelate knowledge items (in the example given, green arrows declare support, red ones declare opposition, whereas the other colors just denote an unjustified relation), another visual cue that appears in Figure 20 concerns the colored rectangles that have been created by participants to cluster related items. Other visual cues bear additional semantics (e.g. the thickness of an edge may express how strongly an item objects another one). The spatial arrangement of the collaboration artifacts aid users have a neat and quick view of the alternative ideas considered so far as well as the underlying argumentation. Since the process of gathering and sharing resources about the particular debate is unstructured, highly dynamic and thus rapidly evolving, this projection provides an appropriate environment to support collaboration at this stage. The aim is to bring the session to a point where main trends crystallize.

Some contexts necessitate a further elaboration of the knowledge items considered so far, and exploitation of additional functionalities such as formal argumentation and reasoning mechanisms. A formal projection in CoPe\_it! is able to cover such needs by providing a fixed set of discourse element and relationship types, with predetermined, system-interpretable semantics. Further elaborating our example, let us assume that the collaboration has reached a point where a switch to a more formal projection is needed. This implies that selected knowledge items' and relationships' types have to be transformed. The above are determined by the underlying visualization and reasoning model of the formal projection (this process can be semi- or fully automated).

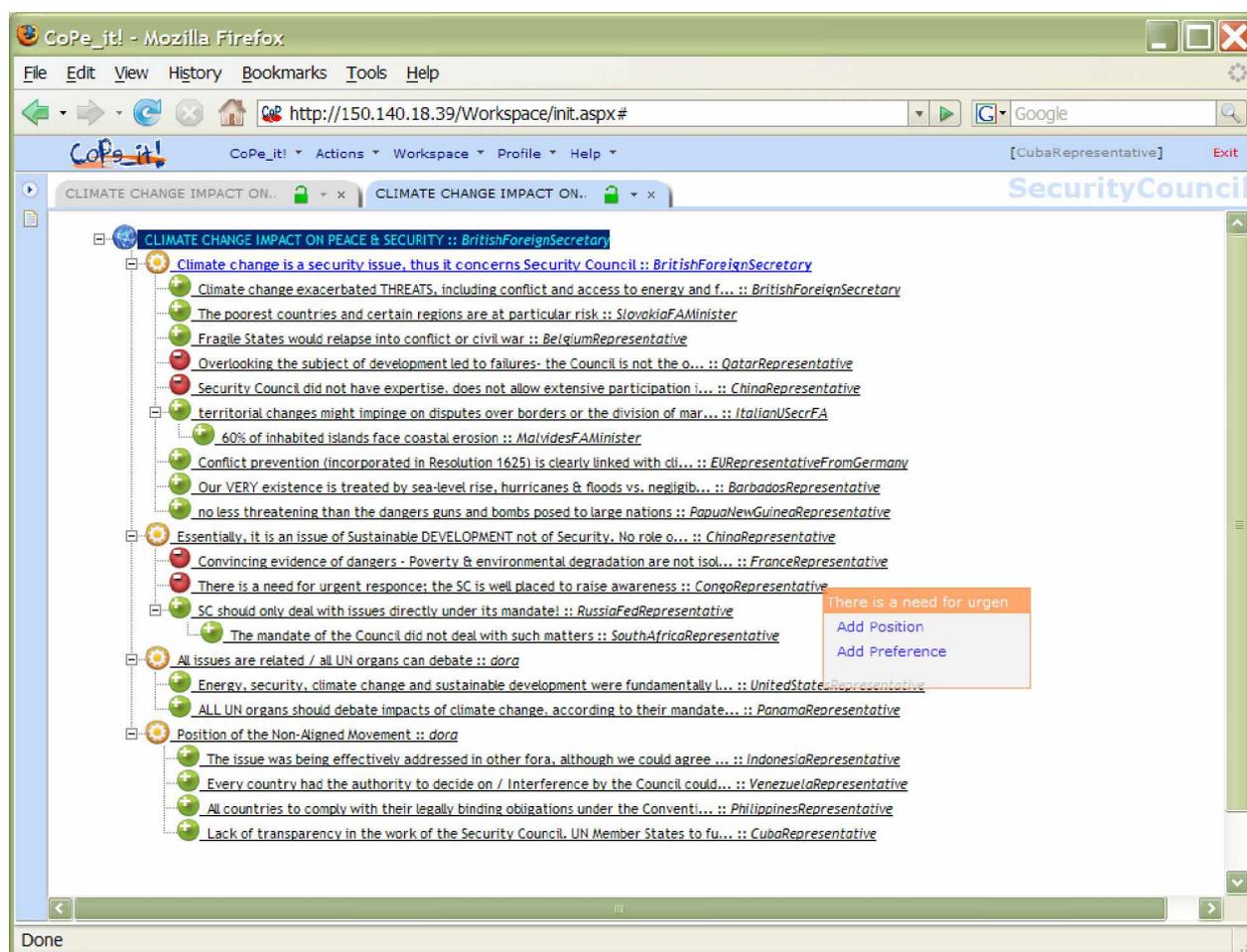


Figure 21: An instance of the formal projection of the collaboration.

An instance of the workspace's formal projection (corresponding to the data-intensive view of Figure 20) is shown in Figure 21. This projection adopts an IBIS-like formalism (item types supported are issues, alternatives and positions – see [Conklin and Begeman, 1989]), and provides a structured language for argumentative discourse together with a mechanism for the evaluation of alternatives. More specifically, the ideas appearing in the informal projection have been transformed to alternatives (alternatives correspond to solutions to the issue under consideration). Other knowledge items have

been transformed to positions in favor or against exploiting the coloring and the legends of the interrelating arrows. Additional reasoning can be performed through the expression of preferences, which provide participants with a qualitative way to weigh reasons for and against the selection of an alternative. Further to the argumentation-based structuring of a collaborative session, this projection integrates a reasoning and scoring mechanism that determines the status of each discourse entry (according to broadly accepted argumentation rules), the ultimate aim being to keep users aware of the most prominent alternative. A detailed description of these mechanisms can be found in [Karacapilidis and Papadias, 2001]. In the instance of Figure 21, the alternative “Climate change is a security issue, thus it concerns Security Council” wins; this can be changed in another instance of the collaboration, depending on the underlying argumentation.

Collaboration through CoPe\_it! may continue at the informal projection, where users are able to exploit a richer set of features and functionalities. Alternative projections of a Collaboration Workspace should be considered jointly, in that a switch from one to the other can further facilitate the argumentative collaboration process. Moreover, a particular collaboration context may be better handled through a less or more formal projection. One may also consider the case where decrease of formality is desirable. For instance, while collaboration proceeds through a formal projection, some discourse elements need to be further justified, refined and elucidated. It is at this point that the collaboration session could switch to a more informal view in order to provide participants with the appropriate environment to better shape their minds.

### 3.2 Using CoPe\_it! in a community of educators

This section presents an example concerning real collaboration between members of a community of educators, aiming on considering alternative teaching modes to (potentially) reach a decision on which is the most appropriate one. Figure 22 illustrates an early instance of the collaborative workspace created for the needs of the above community (for the particular issue under consideration). As shown, only one user has contributed so far (nickname: karakap) by: (i) uploading on the workspace some useful resources (a “very interesting paper” and a “useful URL”), (ii) proposing the idea “traditional lecture” (as an alternative of teaching modes), and (iii) interrelating his idea with two additional items, one that clearly (according to him) argues in favour of the abovementioned proposal (to do so, he has uploaded the argument “The lecture can be used in any size class and is often the only option in large classes”, and has related it - with a green arrow - to the idea “traditional lecture”), and a second one corresponding to related work (“Sullivan and McIntosh paper”, which has been also related to his proposal).

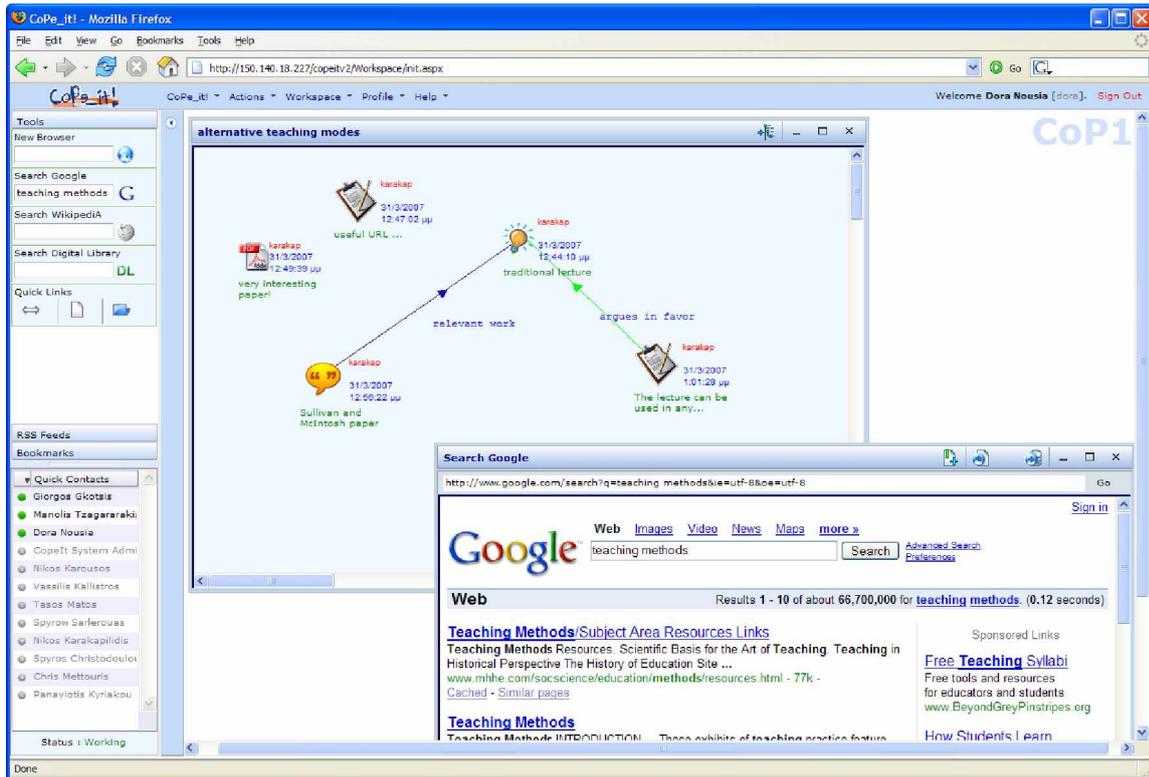


Figure 22: A first instance of the collaborative workspace

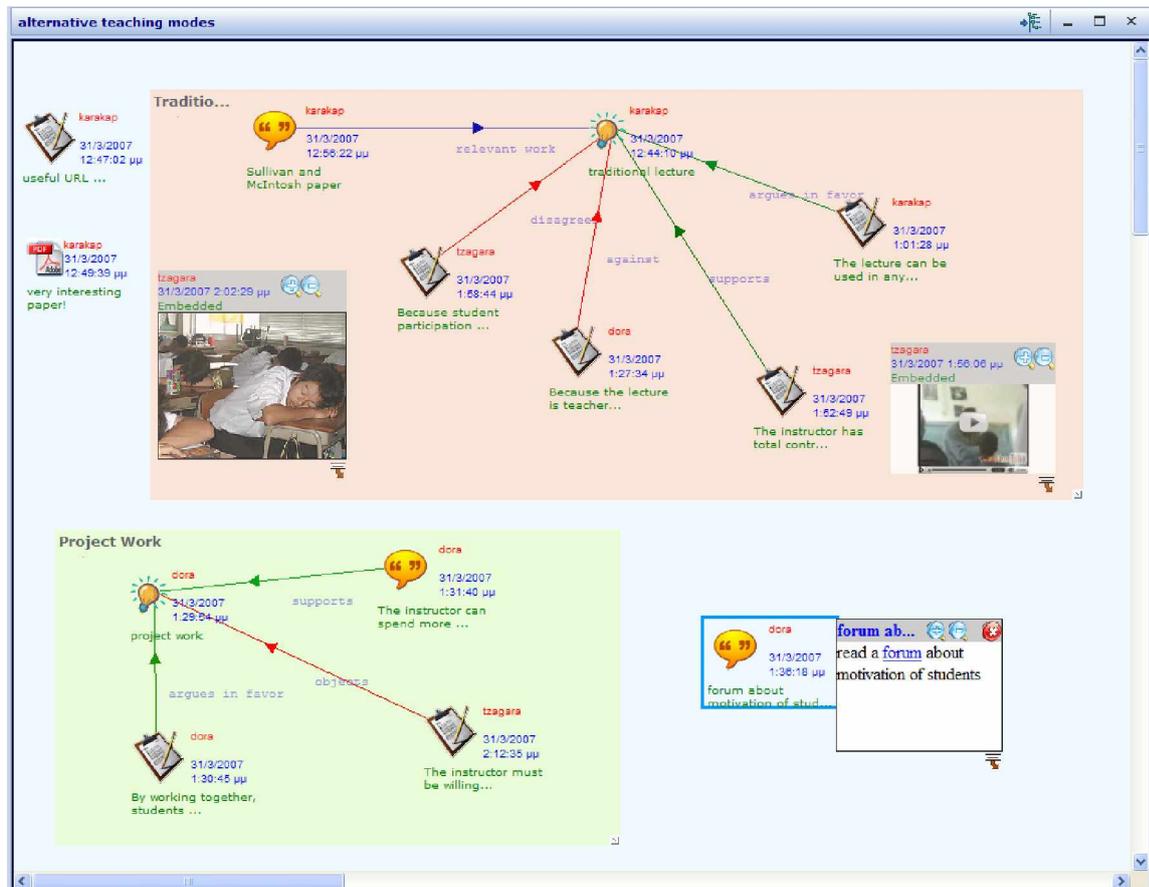


Figure 23: A second instance of the collaborative workspace.

Figure 23 illustrates a second instance of the collaborative workspace under consideration (the screenshot depicts only the workspace area). As shown, two more users (nicknames: dora and tzagara) have been contributed to the collaboration by: (i) proposing a second idea (“project work”, asserted by dora), (ii) uploading additional related resources (e.g. a comment pointing to a “forum about motivation of students”, a comment stating that “The instructor can spend more time with those students or groups who need attention”, a note stating that “Because student participation is minimal, lecturing promotes passivity in students”), (iii) interrelating knowledge items (e.g. the note “By working together, students learn from one another and become less dependent on the instructor” to the idea about “project work”, declaring that the former is an item that “argues in favor” of the latter, or the note “Because the lecture is teacher-centered, it tends to promote one-way communication and the notion that truth resides in the instructor” to the previously asserted idea about “traditional learning”, also declaring that the former is “against” the latter), and (iv) uploading multimedia resources that are relevant to some knowledge items (in the instance shown, a video and an image have been embedded and placed intentionally close to the related items).

Beyond colouring of the arrows that interrelate knowledge items (in the example given, green arrows declare support whereas red ones declare opposition for the specific community), another visual cue that appears in Figure 23 concerns the coloured rectangles that have been created by users to cluster related items (the two rectangles shown correspond to the two alternative ideas proposed so far). Although - at this instance - these rectangles are simply visual conveniences, they may play an important role during the switch to a more formal projection, enabling the implementation of appropriate abstraction mechanisms. Other visual cues supported in this projection may bear additional semantics (e.g. the thickness of an edge may express how strong a resource/idea may object or approve a teaching mode).

Figure 24 illustrates the final state of the collaborative workspace under consideration. As shown, a third idea has come up (“individual instruction”, asserted by tzagara), while additional items have been uploaded and interrelated. The three colour rectangles constructed aid users have a neat and quick view of the alternatives considered as well as the underlying argumentation. Since initially the process of gathering and sharing resources about the available teaching modes is unstructured, highly dynamic and thus rapidly evolving, the projection presented so far provides the most appropriate environment to support collaboration at this stage. The aim is to bring the session to a point where main trends crystallize, thus enabling the switch to a more formal projection (upon the participants’ wish).

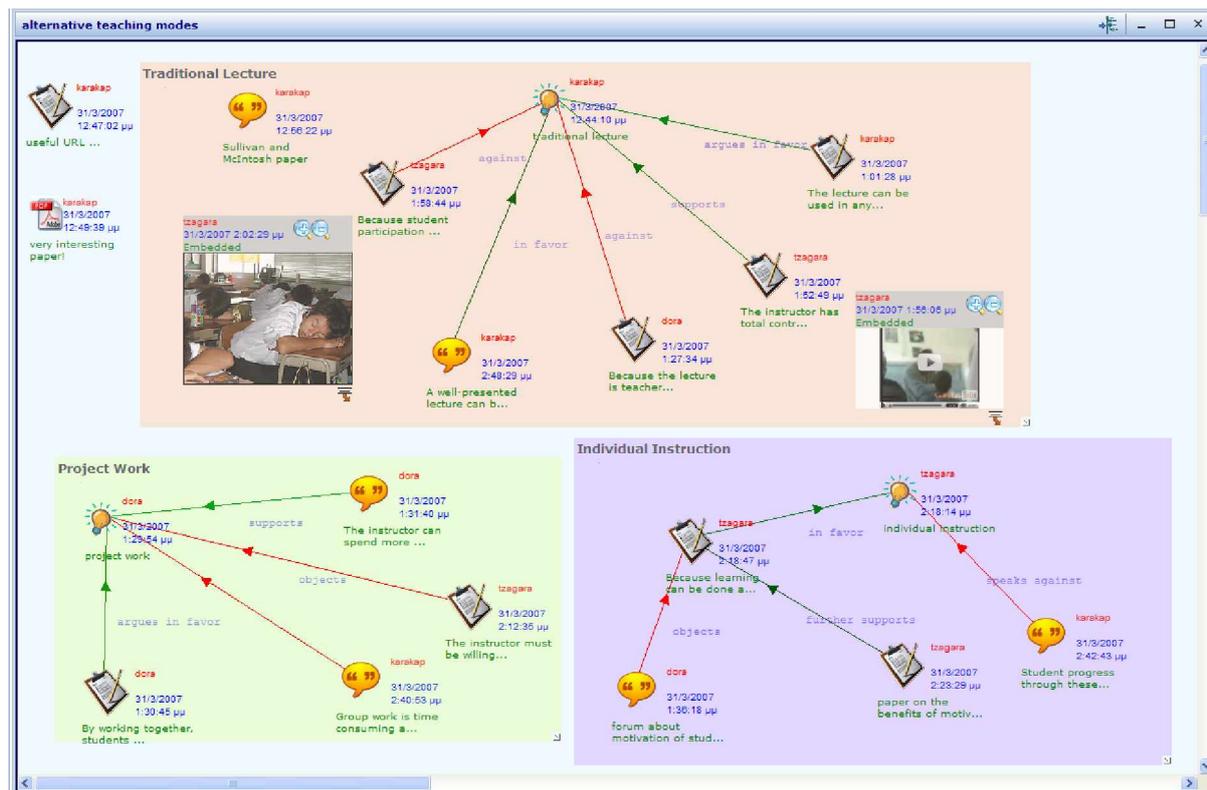


Figure 24: The final state of the collaborative workspace

## Switching Projections

The collaboration instances discussed above correspond to a projection that complies with information triage principles and allows incremental formalization (from a mere collection and sharing of knowledge items to exploitation of legacy resources, interrelation and evolution of knowledge items, and informal/semiformal argumentation and aggregation of knowledge items). Such a projection could perfectly serve the needs of a particular community (for a specific context). However, some communities may have the need to further elaborate the knowledge items considered so far, and exploit additional functionalities to advance their argumentative collaboration. Such functionalities can be provided by other (more formal) projections that may enable the semantic annotation of knowledge items, the formal exploitation of collaboration items patterns, and the deployment of appropriate formal argumentation and reasoning mechanisms. As highlighted above, while an informal projection of the collaboration space aids the exploitation of information by users (user-interpretable view), a formal projection aims mainly at the exploitation of information by the machine (machine-interpretable view). Formal projections provide a fixed set of discourse element and relationship types, with predetermined, system-interpretable semantics.

Further elaborating the example of the previous subsection, let us assume that, at some point of the collaboration, an increase of the formality level is decided (e.g. by an individual user or the session's facilitator). In this case, there is the need to switch to a more formal projection, where knowledge items' and relationships' types have to be transformed, filtered out, or kept "as-is". The above are determined by the underlying visualization and reasoning model of the formal projection (consequently, this process can be partially automated and partially semi-automated). An instance of a projection enabling formal argumentation and group decision making is shown in Figure 25 (the screenshot depicts only the formal projection, which now appears in a separate window; the previous projection is still accessible). This formal projection adopts an IBIS-like formalism [Conklin and Begeman, 1989] and exploits functionalities of a previously developed argumentation support system [Karacapilidis and Papadias, 2001]. It provides a structured language for argumentative discourse and a mechanism for the evaluation of alternatives. Taking into account the input provided by users, this projection constructs an illustrative discourse-based knowledge graph.

The knowledge items allowed in this projection are issues, alternatives, positions, and preferences. Issues correspond to problems to be solved, decisions to be made, or goals to be achieved. For each issue, users may propose alternatives (i.e. solutions to the problem under consideration) that correspond to potential choices. Positions are asserted in order to support the selection of a specific course of action (alternative), or avert the users' interest from it by expressing some objection. A position may also refer to another (previously asserted) position, thus arguing in favour or against it. Finally, preferences provide individuals with a qualitative way to weigh reasons for and against the selection of a certain course of action. A preference is a tuple of the form [position, relation, position], where the relation can be "more important than" or "of equal importance to" or "less important than". The use of preferences results in the assignment of various levels of importance to the alternatives in hand. Like the other discourse elements, they are subject to further argumentative discourse. The above four types of items enable users to contribute their knowledge on the particular problem or need (by entering issues, alternatives and positions) and also to express their relevant values, interests and expectations (by entering positions and preferences). Moreover, the projection continuously processes the elements entered by the users (by triggering its reasoning mechanisms each time a new element is entered in the graph), thus facilitating users to become aware of the elements for which there is (or there is not) sufficient (positive or negative) evidence, and accordingly conduct the discussion in order to reach consensus.

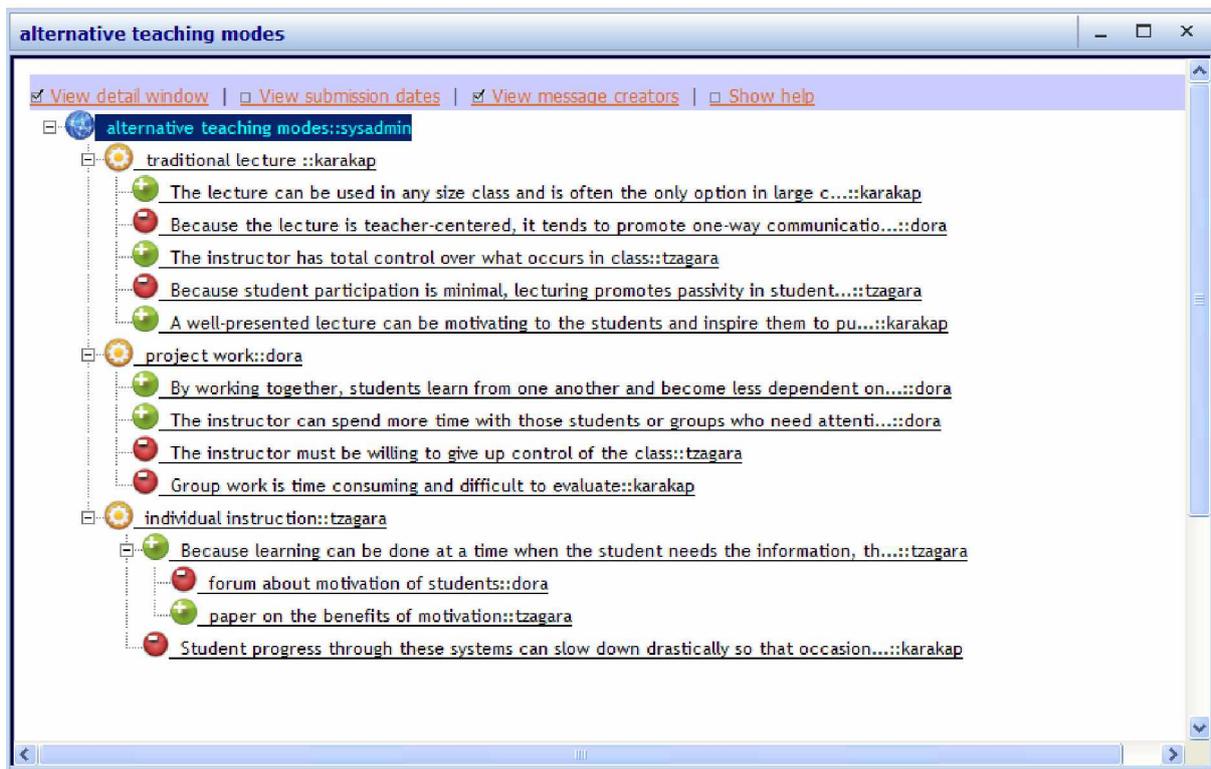


Figure 25: Instance of a more formal projection.

The instance shown in Figure 25 has been automatically built by transforming the projection instance of Figure 24 (the switching to this, more formal, projection has been initiated by the session's facilitator by requesting the related service from the tool). More specifically, the 'ideas' of Figure 24 have been transformed to the 'alternatives' of Figure 25. Other knowledge items have been transformed to positions in favour or against (exploiting the colouring and the legends of the interrelating arrows – for more details see the tools Quick Guide: <http://copeit.cti.gr/site/quickguide.html>).

It is noted that, after the above transformation, the collaboration may continue at this projection, where users are able to exploit a richer set of features and functionalities that is associated to a higher formality level. For instance, further to the argumentation-based structuring of a collaborative session, this projection integrates a reasoning mechanism that determines the status of each discourse entry, the ultimate aim being to keep users aware of the discourse outcome. More specifically, alternatives, positions and preferences of a graph have an activation label (it can be "active" or "inactive") indicating their current status. This label is calculated according to the argumentation underneath and the type of evidence specified for them ("burden of proof"). Activation in this projection is a recursive procedure; a change of the activation label of an element is propagated upwards in the discussion graph. Depending on the status of positions and preferences, the mechanism goes through a scoring procedure for the

alternatives of the issue. At each discussion instance, users are informed about what is the most prominent (according to the underlying argumentation) alternative solution.

Alternative projections of a particular workspace should be considered (and exploited) jointly, in that a switch from one to the other can better facilitate the argumentative collaboration process. One may also consider a particular collaboration case, where decrease of formality is desirable. For instance, while collaboration proceeds through a formal projection, some discourse elements need to be further justified, refined and elucidated. It is at this point that the collaboration session could switch to a more informal view in order to provide participants with the appropriate environment to better shape their minds (before possibly switching back to the formal projection). Switching from a formal to an informal projection is also supported by our approach.

## 4 Future Development Plans

Future development plans cover a series of issues that fall into two categories: (i) issues related to bugs, known limitations of the current release and usability improvements (as reported in D.PAR.04), and (ii) issues related to the integration of new functionality. In the following, we outline the development plans with respect to the latter.

### *Search related*

- Currently, CoPe\_it! permits simple keyword search that is restricted to the collaboration object type (only workspace items) and takes into consideration only one of their properties (their title). Future search functionality will extend the set of types supported, including workspaces and users. Moreover, formulating queries with multiple criteria taking into consideration more object properties (such as last access time or creator) will be possible.
- Future versions of the search functionality will take into consideration the semantic dimension of all CoPe\_it! objects by exploiting the assigned tags that relate to ontologies, as well as the work being performed in WP5 with respect to “global search” scenario.

### *Workspace related*

- Additional Views and Projections which will elaborate various visualization issues are foreseen. For instance, we foresee the implementation of a filtered view of workspaces aiding the appropriate “cleaning” of data intensive workspaces (other mechanisms will be also elaborated).
- Currently, when users work in formal projections of workspaces, changes are not reflected in the informal ones; effort will be put in overcoming related problems in order to synchronize the views provided.
- Currently, collaboration in workspaces is asynchronous. Future releases will provide support for synchronous collaboration within workspaces either in the form of concurrent modifications perceivable in real-time by all connected users or in the form of chat functionality.
- Extending the set of available reasoning mechanisms, which will be available in the formal projections, is also planned. Depending on the nature of the decision making process, CoPs may then select the appropriate one.

- Work towards demonstrating (through animation) the evolution of a workspace to support historical awareness, in order to overcome problems faced by newcomers or absent CoP members.
- Work towards extension of the supported abstraction mechanisms (e.g. aggregating and collapsing a cluster of items, dragging of a cluster of items, etc.).

#### *Personalization related*

- Although the current release provides a number of personalization services, future development steps aim at extending this set of services focusing on awareness and notification issues.

#### *Social space related*

- Work towards providing functionalities that allow a better coordination and monitoring of a CoP's actions (e.g. allowing a CoP coordinator to schedule specific activities, allowing a CoP member to view summaries of activities).
- Consideration of integrating additional functionality to better support socializing.

#### *Interoperability related*

- In order to seemingly interoperate with other tools in the Palette project, internal aspects of CoPe\_it! need to be adapted to facilitate interoperability efforts.
- Integration with selected KM services. The overall goal of this integration effort is to augment existing services of CoPe\_it! by utilizing ontologies that have been developed in the context of Palette's knowledge management services (work carried out in WP3). The use of such ontologies will support the semantic dimension of CoPe\_it! services, thus improving their quality.

An issue which is also going to be thoroughly addressed concerns the overall visualization scheme of the tool. The colors, fonts, themes, icons, logos used in the UI are going to be reconsidered and redesigned in order to achieve uniformity and enhance user experience (this also implies collaboration with work being performed in the context of WP5).

The abovementioned development plans are the outcome of an ongoing brainstorming process within the CoPe\_it! development team, which takes into account the Palette project's participatory design

approach. The priorities regarding their implementation are defined and revised through this very process.

## 5 References

Conklin, J., Begeman, M. (1989). gIBIS: A tool for all reasons, *Journal of the American Society for Information Science* 40(3), pp. 200-213.

Ferraiolo, D.F et al (2001). Proposed NIST Standard for Role-Based Access Control. *ACM Transactions on Information and System Security*, Vol. 4, No 3, pp 224-274.

Karacapilidis, N., Papadias, D. (2001). Computer Supported Argumentation and Collaborative Decision Making: The HERMES system. *Information Systems*, Vol. 26, No 4, 2001, pp. 259-277.

## Appendix A: Some figures about the tool's active use

For the period 01/08/2007 - 31/01/2008:

- One hundred and thirty three (133) different users from 364 different IPs have participated in CoPe\_it!.
- Total actions (create, open, move, update, link, or delete a collaboration object) performed in the Desktop View Collaboration Workspaces are 21,844.
- Three hundred ninety six (396) Collaboration Workspaces have been totally created (some of them have been deleted), while 1,519 items have been uploaded on these spaces.
- Total active Collaboration Workspaces are three hundred forty six (346), out of which two hundred forty three (243) are private and one hundred and three (103) are public.
- Twenty two (22) communities have been created, namely: "Test Community" (20<sup>‡</sup>), "Earth" (202), "learn-nett" (13), "TeL" (9), "Teachers in the four cardinal points" (15), "Real-E-Learning" (2), "Curriculum M&T" (6), "Greenhouses" (9), "Malengrez's Familiy" (1), "Community Deborah" (2), "teimes" (24), "sxolia" (1), "testing cop-it!" (4), "LucoSpace" (2), "Mike com" (1), "My team" (4), "CopeitCom" (1), "ELO COM" (1), "accompagnement" (1), "Team Lyon" (1), "Soleil Vert" (4), "Scenarisation Did@cTIC" (4).

---

<sup>‡</sup> The numbers in the parentheses represent the number of members of each community; many members are currently participating in more than one communities

## Appendix B: License & Code information (updated)

CoPe\_it! is a web based tool. Its current version (v2.2) is provided to everyone for free and “as is” for evaluation and experimentation purposes. The copyright owner (©2007 CTI / eLearning and Social Software Sector) has the right to change it without any prior notification to the users. The owner of the copyright is not liable for any damages including loss of information related to, or arising out of its use. The source code of CoPe\_it! can be provided through a CVS (Concurrent Versions System).

CoPe\_it! is built using mostly C# code for the application logic and Javascript for the front-end. The list of SW needed for a new installation of a CoPe\_it! server includes the following:

- Internet Information Server (IIS)  
Purpose: Web server hosting CoPe\_it! Web application  
Licensing: proprietary (comes with Windows Operating systems. For non-server versions, the number of sessions is limited to 10).
- Microsoft .NET Framework 2.0.  
Needed for executing the CoPe\_it! Web application.  
Purpose: run/execute application logic (written in C#)  
Licensing: Freeware (free download, install and deployment of the framework for commercial or non commercial uses).
- Microsoft SQL Server 2005  
Purpose: database back-end.  
Licensing: Commercial with various licensing options (per CPU etc).  
Installation of Microsoft SQL Server 2005 requires clients to purchase licenses. The SQL Server 2005 Runtime permits redistribution of SQL Server 2005 along with CoPe\_it! to clients without obligating the client to purchase licenses. In this case, the SQL Server 2005 Runtime is permitted to be used only by the application (CoPe\_it!). The client of the application cannot use this SQL Server product to run other applications or to develop new applications, databases, or tables. The vendor (CTI is vendor for the CoPe\_it! current installation) have a Product Integration Agreement so that it is able to redistribute the SQL Server 2005 Runtime. For more information see:  
<http://support.microsoft.com/kb/917400>
- TIBCO General Interface v3.4.1  
([http://www.tibco.com/software/rich\\_internet\\_application/general\\_interface/default.jsp](http://www.tibco.com/software/rich_internet_application/general_interface/default.jsp))  
Purpose: javascript library  
Licensing: Open source BSD License (<http://www.opensource.org/licenses/bsdlicense.php>)
- Script.aculo.us v1.7.0 (<http://script.aculo.us>)  
Purpose: javascript library(/  
Licensing: MIT license. Allows reuse within proprietary applications, as well as in free and open source software (<http://wiki.script.aculo.us/scriptaculous/show/License>).

- SharpZipLib  
Purpose: .Net library for zip and unzip operations  
Licensing: The library is released under the GPL.  
(<http://www.icsharpcode.net/OpenSource/SharpZipLib/>)
- URL2JPEG v1.1 (<http://www.url2jpeg.com>)  
Purpose: Easy web page capture for thumbnail creation.  
Licensing: Available under the BSD License
- ExplorerCanvas v.2 (<http://sourceforge.net/projects/excanvas/>)  
Purpose: javascript library, allowing 2D command-based drawing  
Licensing: Apache License, Version 2.0 (<http://www.apache.org/licenses/LICENSE-2.0>)
- Open-jACOB Draw2D v.2.36 (<http://www.openjacob.org>)  
Purpose: javascript library, allowing 2D command-based drawing  
Licensing: GNU Lesser General Public License (LGPL) (<http://www.gnu.org/licenses/lgpl.html>)
- FCK Editor v.2.5 (<http://www.fckeditor.net>)  
Purpose: HTML text editor  
Licensing: Distributed under the [GPL](#), [LGPL](#) and [MPL](#) open source licenses.
- ThumbnailList generator  
Purpose: custom win32 application; filters the entire list of CoPe\_it! Collaboration Workspaces and generates an XML list of thumbnails that have been created or modified in a particular period of time.

## Appendix C: Google Analytics

CoPe\_it! has been registered with “Google Analytics” (<http://www.google.com/analytics>), a free web site statistic tool that provides useful information about how many people visited a site, how they found it, how they explored it, etc. (Figure 26).

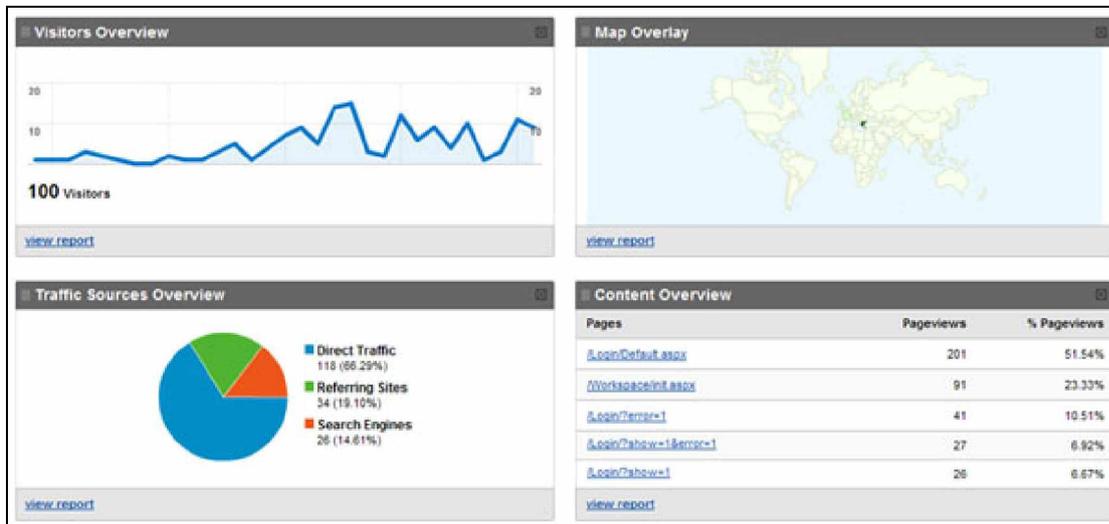


Figure 26: Overview of the Google Analytics report for <http://copeit.cti.gr>

In particular, whenever a user visits CoPe\_it!, Google Analytics tracks down this event. With this information, it is possible to extract useful information, including:

- § Visitors' information
  - Total visits per period of time
  - Unique visitors
  - Time on Site
  - Visitors' Countries and Languages
  - Browser Capabilities
- § Traffic Sources
  - All different traffic sources
  - Search Engines and Keywords
  - Referring Sites

This information will be taken into account by the CoPe\_it! development team to better shape particular development plans (e.g. issues related to browser compatibility, localization, users engagement etc.).