Fostering creativity in learning through digital games

C²Learn
creative learning

Creative Emotional Reasoning Computational Tools Fostering Co-Creativity in Learning Processes

www.c2learn.eu

SEMANTIC REASONING COMPUTATIONAL TOOLS

C²LEARN PROJECT DELIVERABLE NO. D3.1.2C

Authors: National Centre for Scientific Research "Demokritos" (NCSR-D)

Dissemination level: Public

DOCUMENT IDENTITY

Project category	Details
Deliverable code	D3.1.2b
Full title	Semantic Reasoning Computational Tools
Work package	WP3
Task	T3.1 Semantic Reasoning
Consortium partners leading	NCSR-D
Consortium partners contributing	NCSR-D

DOCUMENT HISTORY

Version	Date	Handling partner	Description
v0.1	31/09/2013	NCSR-D	Improvements on Content Safety
v0.2	31/10/2013	NCSR-D	Extensions of Content Safety mechanics
v0.3	15/11/2013	NCSR-D	Document Setup, ToC
v0.4	25/11/2013	NCSR-D	Release of CTP Android
v0.5	31/11/2013	NCSR-D	Document updates / revisions
V0.7	31/01/2014	NCSR-D	Tools assessing Creativity
V0.8	28/02/2014	NCSR-D	Tools Updates / Revision
V0.9	15/03/2014	NCSR-D	Document updates / revisions
V1.0	01/04/2014	NCSR-D	Final version of C²Learn Semantic Reasoning Computational Tools
V1.1	12/09/2014	NCSR-D	Version (b) including updates/revisions of the services and refinement of the link to CPS (cf. D3.4.2)
V1.3	15/10/2014	NCSR-D	Addition of Semantic Distance Calculation Service
V1.5	30/10/2014	NCSR-D	Version (c) including Improvements on Services Performance

EXECUTIVE SUMMARY

C^2Learn at a glance

C²Learn (www.c2learn.eu) is a three-year research project supported by the European Commission through the Seventh Framework Programme (FP7), in the theme of Information and Communications Technologies (ICT) and particularly in the area of Technology-Enhanced Learning (TEL) (FP7 grant agreement no 318480). The project started on 1st November 2012 with the aim to shed new light on, and propose and test concrete ways in which our current understanding of creativity in education and creative thinking, on the one hand, and technology-enhanced learning tools and digital games, on the other hand, can be fruitfully combined to provide young learners and their teachers with innovative opportunities for creative learning. The project designs an innovative digital gaming and social networking environment incorporating diverse computational tools, the use of which can foster co-creativity in learning processes in the context of both formal and informal educational settings. The C²Learn environment is envisioned as an open-world 'sandbox' (non-linear) virtual space enabling learners to freely explore ideas, concepts, and the shared knowledge available on the semantic web and the communities that they are part of. This innovation is co-designed, implemented and tested in systematic interaction and exchange with stakeholders following participatory design and participative evaluation principles. This happens in and around school communities covering a learner age spectrum from 10 to 18+ years.

About this document

Deliverable D3.1.2c is an update on the previous version of the Semantic Reasoning Computational Tools deliverable (D3.1.2b). The document provides information on the different types of computational tools, presents the improvements and extensions concerning the performance of the services implementing the tools, and demonstrates their usage within a C2Learn Game.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF TERMS AND ABBREVIATIONS

Term/Abbreviation	Definition
CER	Creative Emotional Reasoning
LT	Lateral Thinking
LTC^2	C^{2}Learn's Lateral Thinking
SLT	Semantic Lateral Thinking
TeL	Technology-Enhanced Learning

1. INTRODUCTION

C2Learn's Lateral Thinking (LTC2) theory is built on Lateral Thinking and Brainstorming Techniques. The term Lateral Thinking (LT) is invented in 1967 by Edward De Bono [1]. As he elucidates:

"The brain as a self-organizing information system forms asymmetric patterns. In such systems there is a mathematical need for moving across patterns. The tools and processes of lateral thinking are designed to achieve such 'lateral' movement. The tools are based on an understanding of self-organizing information systems."

"In any self-organizing system there is a need to escape from a local optimum in order to move towards a more global optimum. The techniques of lateral thinking, such as provocation, are designed to help that change."

In deliverable D2.1.x (Creative Emotional Reasoning), three kinds of LT are suggested: Semantic, Diagrammatic, and Emotive. In the context of the present deliverable, we elaborate on the Computational Tools related to Semantic Lateral Thinking.

2. SEMANTIC REASONING COMPUTATIONAL TOOLS

This section discusses the rationale behind the design and implementation of the C2Learn Semantic Reasoning Computational Tools, with respect to the underlying theories for CER. A detailed description of the distinct tools follows.

2.1 THEORETICAL BACKGROUND

In deliverable D2.1.x (Creative Emotional Reasoning), three kinds of LT are suggested: Semantic, Diagrammatic, and Emotive. In the context of the present deliverable, we elaborate on the Computational Tools related to the first of these kinds of LT, i.e. Semantic Lateral Thinking. Semantic Lateral Thinking (SLT) involves the use of different conceptual Po1 [2], a tool or an operator meant to provoke and dislocate from habitual patterns and forms, as well as disassociate established connections. SLT is divided into four techniques, namely, Random Stimulus, Re-conceptualization, Escapism, and Role-play.

The following subsections provide a brief analysis of these techniques with respect to the process of identifying the type of computational tools needed to support them within a digital environment. For further details on the techniques, cf. the aforementioned deliverable D2.1.1.

2.1.1 RANDOM STIMULUS

The main principle of this SLT technique is the introduction of a foreign conceptual element with the purpose of disrupting preconceived notions and habitual patterns of thought. The user/participant is thus forced to integrate/exploit the foreign element in the production of a solution/idea, and bring together disparate domains.

---

[1] Po is a term coined by Edward de Bono and connotes an idea which opens up a space for new ideas and solutions to develop. It is extracted from words as suppose, hypothesis, potential, possibility, and poetry.

Presentation of Problem/Task

↓

Introduction of **Random Stimulus**

↓

Discussion/Brainstorming for New Solution/Idea

↓

Presentation of **New Solution/Idea**

↓

Debating on New Solution/Idea
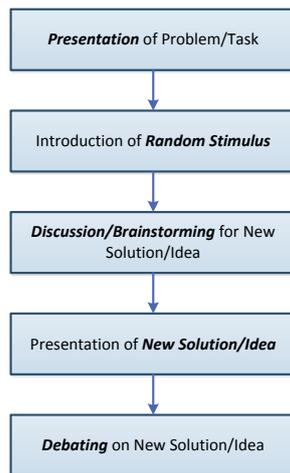
Randomness is the main guarantor of foreignness and hence of stimulation of creativity. Foreignness in this context has two main dimensions: (a) It is important that the player feels that he/she has to somehow integrate/exploit an element which is introduced completely from without, whose introduction is in no way under the player's control. In some ways an intruder has to be re-conceptualized as a friendly aid; and (b) the new element should, at least initially, be as unconnected as possible to the subject/type/structure of the problem. By doing so, we someway ensure that no unconscious/unobserved pre-established analogies, preferences and connections creep in the selection of the stimulus.
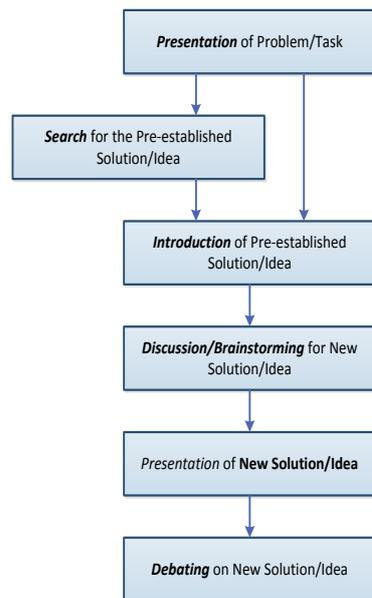
After the presentation of the problem, the participants are asked to use creatively in the reasoning process the random stimulus provided. An example of the workflow alongside is presented in Table 1.

In this table, the activities performed by the respective actors (educator, learner, group) are indicated, and a brief description of these activities is provided, along with a relevant example. Moreover, each activity is characterized based on the DialogPlus Learning Taxonomy [3].

Learning Activity	Description	Example	DialogPlus Learning Taxonomy		
			Technique (How)	Interaction (Who)	Roles (Which)
Presentation of *Problem/Task*	The problem or task at hand is presented	"A mountainous village is frequently ravaged by heavy snowstorms. Devise ways to deal with this."	Presentation	One to Many	Moderator Group Leader
Introduction of **Random Stimulus**	The stimulus is presented and assimilated by the participant(s).	We get **poetry** as a random word. The participant(s) is thus forced to integrate/exploit the foreign element in the production of a solution.	Assignment	Group-based	Group Participant(s)
Discussion/Brainstorming for New Solution/Idea	The production of a **bridging idea** falls within this **iterative** activity, consisting of three time-constrained phases: Communication, Production/Adaptation, and Information Handling, which are repeated as necessary.	The bridging idea could be that **a poem has structure**. The notion of **structure** can then lead us down different paths, which we may later combine to advance our research. e.g. We may view the natural event as **a structured occurrence**, and thus one that may be studied methodically. We then begin by examining the patterns inherent in it, in the hopes of finding useful data that will help us predict future snowstorms and more effectively deal with their consequences.	Brainstorming	Group-based	Group Participant(s)
Presentation of **New Solution/Idea**	The outcome of the previous phase is presented to all participants.	By combining the produced ideas/ solutions the inherent properties of the problem can be revealed i.e. that the problem at hand concerns **a structured event, calling for a structured and possibly structural solution**.	Presentation	Class-based	Rapporteur(s)
Debating on New Solution/Idea	Debate on the effectiveness of the solution/idea produced. In the case that an idea is not well grounded, the whole (or part of the) process may be repeated	The solution emerged appears to be a concise view of the matter. What's more, this established idea can then transferred onto other catastrophic natural phenomena such as earthquakes, floods etc.	Arguing/ Debate	Class-based	Moderator Individual Learner

Table 1: Random Stimulus Workflow Example

2.1.2 RE-CONCEPTUALIZATION

```
Presentation of Problem/Task
        ↓
Search for the Pre-established
Solution/Idea
        ↓
Introduction of Pre-established
Solution/Idea
        ↓
Discussion/Brainstorming for New
Solution/Idea
        ↓
Presentation of New Solution/Idea
        ↓
Debating on New Solution/Idea
```
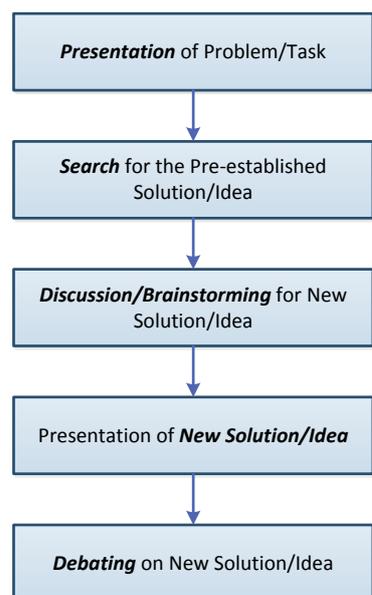
Re-conceptualization involves the use of already established solutions and ideas in new environments. One is encouraged in exploiting the potential of familiarity in the production of novel ideas. The familiar features of the established solution/ idea will re-inscribe themselves on the unfamiliar environment or appear in a new light.

An example of the workflow alongside is presented in Table 2.

In this table, the activities performed by the respective actors (educator, learner, group) are indicated, and a brief description of these activities is provided, along with a relevant example. Moreover, each activity is characterized based on the DialogPlus Learning Taxonomy.

2.1.3 ESCAPISM

```
Presentation of Problem/Task
        ↓
Search for the Pre-established
Solution/Idea
        ↓
Discussion/Brainstorming for New
Solution/Idea
        ↓
Presentation of New Solution/Idea
        ↓
Debating on New Solution/Idea
```

Escapism is a general term for the use of what-if counterfactuals in the production of new solutions/ ideas. This temporary emancipation from the given, which is possible in a variety of forms, will not only facilitate the production of new and unusual ideas, but will also localize and bring more into focus the conditions within which the problem is situated.

An example of the workflow alongside is presented in Table 3.

In this table, the activities performed by the respective actors (educator, learner, group) are indicated, and a brief description of these activities is provided, along with a relevant example. Moreover, each activity is characterized based on the DialogPlus Learning Taxonomy.

2.1.4 ROLE-PLAY

Role-play is a general term for the use of as-if counterfactuals in the production of new solutions/ ideas. The technique builds upon the everyday experience/ intuition of a perspective. A change of perspective is meant to reorient as within a given problem, revealing new paths to be explored, but also to make us aware of unobservable limitations in our habitual way of seeing things.

As Role Play is a highly complex technique, with different parameters that are dependent on the participants' personality, their perception of roles and their familiarity with the role characteristics, it is not trivial to define a concrete, standard workflow for the activity. Thus, such a workflow has not been defined in D2.1.1. However, it is possible to use flexible workflow definitions, based on the concepts and actions of the other, aforementioned techniques.

Learning Activity	Description	Example	DialogPlus Learning Taxonomy		
			Technique *(How)*	Interaction *(Who)*	Roles *(Which)*
***Presentation** of Problem/Task*	The problem or task at hand is presented.	"A mountainous village is frequently ravaged by heavy snowstorms. Devise ways to deal with this."	Presentation	One to Many	Moderator Group Leader
***Search** for the Pre-established Solution/Idea (optional step)*	Search available resources/ ideas on the given problem/ task.	In searching for an established solution we come across the idea of a **greenhouse**, as well as, the idea of a **smart house**.	WEB Search	Group-based	Group Participant(s)
***Introduction** of Pre-established Solution/Idea*	Debate/ argue so as to select the dominant idea to work with. Familiarize with the selected solution/idea.	The idea of the greenhouse was selected, and the references/ resources describing greenhouses were presented and discussed within the class.	Arguing/ Debate	Class-based	Moderator Individual Learner
			Presentation	One to Many	Moderator Group Leader
***Discussion/Brainstorming** for New Solution/Idea*	The process of ***conceptual transformations*** and ***exploration of extensions*** for the production of ***links*** falls within this activity.	What is constitutive of a greenhouse as a solution is the idea of regulating the environmental conditions within it. Our task now is to create sufficient links that will turn this idea into a crisis-solver and possibly beyond.	Brainstorming	Group-based	Group Participant(s)
*Presentation of **New Solution/Idea***	The outcome of the previous phase is presented to all participants.	By combining the produced ideas/ solutions the inherent properties of the problem can be revealed.	Presentation	Class-based	Rapporteur(s)
***Debating** on New Solution/Idea*	Debate on the effectiveness of the solution/idea produced. In the case that an idea is not well grounded, the whole (or part of the) process may be repeated	The solution emerged appears to be a concise view of the problem in hand.	Arguing/ Debate	Class-based	Moderator Individual Learner

Table 2: Re-conceptualization Workflow Example

Learning Activity	Description	Example	DialogPlus Learning Taxonomy		
			Technique *(How)*	Interaction *(Who)*	Roles *(Which)*
Presentation *of Problem/Task*	The problem or task at hand is presented.	"A mountainous village is frequently ravaged by heavy snowstorms. Devise ways to deal with this."	Presentation	One to Many	Moderator Group Leader
Search *for the Pre-established Solution/Idea*	Introduction of a ***'what if'*** ***counterfactual***.	We begin by posing an escapist type of counterfactual, for example: **What if we could control the weather?**	Assignment	Group-based	Group Participant(s)
Discussion/Brainstorming *for New Solution/Idea*	The production of a ***provisional idea*** falls within this activity, occurring during its initial cycles.	… the solution in this hypothesis seems clear: Control over the weather means control over the phenomenon (snowstorms in this case), which means the crisis can be easily dealt with. The participants discuss / produce ideas/ solutions on "how are we to use this?".	Brainstorming	Group-based	Group Participant(s)
Presentation of ***New Solution/Idea***	The outcome of the previous phase is presented to all participants.	By combining the produced ideas/ solutions the inherent properties of the problem can be revealed.	Presentation	Class-based	Rapporteur(s)
Debating *on New Solution/Idea*	Debate on the effectiveness of the solution/idea produced. In the case that an idea is not well grounded, the whole (or part of the) process may be repeated	The solution emerged appears to be a concise view of the problem in hand.	Arguing/ Debate	Class-based	Moderator Individual Learner

Table 3: Escapism Workflow Example

2.2   COMPUTATIONAL TOOLS FOR SUPPORTING SLT TECHNIQUES

The following table summarizes the Computational Tools that were designed and implemented as assistive technologies for the Semantic Lateral Thinking techniques presented in the previous section. There are four (4) tools that are actively integrated in the process of creative thinking with the purpose of fostering idea conception and expression; the Thinking Seeds Generator, the Web Miner, the Cloud of Thoughts, and the Competitive Thinking Spaces tools. Additionally, the set of Computational tools includes the Creativity Points Computation service, a tool for assessing creativity over text segments. Finally, two supporting tools, used by the other services, have also been implemented, the Search Engine Wrapper and the Text Clustering Service.

Type	Tool
Tools fostering Idea Conception	Thinking Seeds Generator
	Web Miner
	Cloud of Thoughts
	Competitive Thinking Spaces
	Semantic Distance Calculation
Tools for assessing dimensions / aspects of Creativity	Creativity Points Computation
Supporting Tools	Search Engine Wrapper
	Text Clustering

Table 4: C2Learn Semantic Reasoning Computational Tools

The following subsections present in detail the different Semantic Reasoning Computational tools, describing their functionality and providing details on the methodology and implementation of each tool.

2.3   SEMANTIC REASONING COMPUTATIONAL TOOLS FOSTERING IDEA CONCEPTION

This section provides details on the design and implementation of the C²Learn Semantic Reasoning Computational Tools for Fostering Idea Conception. All the tools are implemented as SOAP-based web services, and thus are able to be used by any client that follows the services' call definition. The WSDL definition of the C2Learn Semantic Reasoning Computational Tools can be found at: *http://cru.iit.demokritos.gr:8080/c2Learn/C2learn?wsdl*

2.3.1   THINKING SEEDS GENERATOR

2.3.1.1   FUNCTIONALITY

The Thinking Seeds Generator provides a textual stimulus, having a varying semantic distance from its input. The produced word, as it is semantically distant from the initial state, is meant to act as an initiative to think out-of-the-box, re-contextualize ideas or be led to examine other perspectives of a problem / situation.

*The tool supports input in all three languages covered by C2Learn (**English, German,** and **Greek**).*

2.3.1.2   IMPLEMENTATION

The input of the Thinking Seeds Generator is a seed phrase in any of the three supported languages (English, German, and Greek) and the degree of difficulty, which denotes the semantic distance between the random words that will be returned and the initial phrase. In this context, the semantic distance of two terms is the number of edges in the WordNet [4] synset graph that must be traversed in order to reach to a synset starting from another- specific- synset.

The initial word to be used for the process is determined depending on the size of the textual input. When the input is a single word or a phrase up to three words, the input is processed as it is. In the case of larger texts, the service discovers the dominant terms within the text as follows:

1. Stopwords are removed from the text;
2. The remaining words are stemmed, and hashed with respect to their stem;
3. The three (3) most frequent stems are considered as dominant and the words having these stems are considered the dominant words within the text;
4. One of the dominant words is selected randomly as the seed to be used.

Following the process of determining the seed word, the service traverses the WordNet graph according to the following methodology:

1. Retrieve the WordNet synsets to which the seed word belongs;
2. Start from the synset containing the most words;
3. Select the word in the selected synset that belongs to the most synsets;
4. Repeat the first two steps for the selected word until the number of steps is equal to the set difficulty;
5. Select all the words belonging to the last visited synset.

2.3.2   WEB MINER

2.3.2.1   FUNCTIONALITY

The Web Miner is used to provide a summary of web content that is related to an input text segment of variable size. The summary is expressed as a tag cloud structure, i.e. the service returns a set of dominant words found in the examined web content, along with their frequency of appearance.

*The tool supports input in all three languages covered by C2Learn (**English**, **German**, and **Greek**).*

2.3.2.2   IMPLEMENTATION

The input of the service is a word or short phrase, and an indicator that specifies if the service should only handle content safe for children. The service invokes the Search Engine wrapper (see Section 2.5.1) and retrieves the HTML content of the first 50 results. The content is cleaned using the boilerpipe library [5] in order to obtain the textual content of these pages. The stopwords present are removed and the remaining content is stemmed –using the Snowball stemmer [6] - and hashed in order to calculate the TF-IDF value for each distinct stem.

Let *P* the number of pages returned from the Search Engine Wrapper, p an individual page in *P* and t a distinct stem. The weighted frequency of t in P is:

$$WF(t,P) = \frac{\sum_{p \in P} tf(t,p) \times idf(t,P)}{|P|}$$

where *tf* is the logarithmically scaled term frequency of term t in a page p

$$tf(t,p) = \log(f(t,p) + 1)$$

and *idf* is the inverse document frequency of this term in set *P*

$$idf(t,P) = \log \frac{|P|}{|\{p \in P : t \in p\}|}$$

For each distinct stem, the most frequent form (with respect to its raw number of occurrences) is chosen to build a structure that encapsulates the {stem form, weighted frequency} pairs for the entire content.

It is important to note that the Web Miner service does not pose any limitations on the visualization of the tag cloud. The returned structure defines just the most frequent terms found in the relevant web pages and their respective frequency. It is, therefore, left to the client application to incorporate the tag-cloud in its environment, using the most suitable representation.

2.3.3   CLOUD OF THOUGHTS

2.3.3.1   FUNCTIONALITY

The *Cloud of Thoughts* service provides a summary of a text segment, by examining the dominant words / short phrases found within the segment and returning them as a tag cloud structure. Its aim is to identify and present the major ideas present in the text, giving to the user a synopsis of others' thoughts that can lead him / her to change thinking perspectives, guiding his thought in a different path.

*The tool supports input in all three languages covered by C2Learn (**English**, **German**, and **Greek**).*

2.3.3.2   IMPLEMENTATION

The service is invoked with the text to be summarized as its input. After the removal of stopwords, it calculates the logarithmically scaled term frequency as shown above. Finally, a structure that encapsulates the {stem form, term frequency} pairs is returned.

2.3.4   COMPETITIVE THINKING SPACES

2.3.4.1   FUNCTIONALITY

The Competitive Thinking Spaces service relies on the premise that a text segment may contain different aspects / points of view and the user can focus on a specific one to proceed with a line of thought. Thus, the service analyses a text fragment and identifies different groupings of the concepts included in the fragment, returning them to the caller.

*The tool supports input in all three languages covered by C2Learn (**English**, **German**, and **Greek**).*

2.3.4.2   IMPLEMENTATION

In order to determine the thinking spaces, the service operates on a text segment provided as input. It discards stopwords and then employs the Text Clustering Service (see Section 2.5.2) in order to cluster the word set. If the produced clusters exceed a specified number (e.g. 4), the service reduces the clusters to this number, using the distances between the clusters. It finally returns to the calling agent a structure that encapsulates the clusters and the words / phrases belonging to each cluster.

2.3.5   SEMANTIC DISTANCE CALCULATION

2.3.5.1   FUNCTIONALITY

The Semantic Distance Calculation Service operates on pairs of words and returns their semantic distance, i.e. a metric for their disassociation semantically.

*The tool supports input in all three languages covered by C2Learn (**English**, **German**, and **Greek**).*

2.3.5.2 IMPLEMENTATION

The semantic distance between two words is calculated based on their morphological similarity and their thematic correlation. Thus, the employed metric is a combination of a string distance measure and a WordNet distance measure. More specifically, we use the Levenshtein distance for measuring the similarity of the two words as strings, and a WordNet distance for measuring their semantic association. The WordNet distance is obtained via the following algorithm:

```
Input: Two words w1 and w2
Identify the first common parent P of w1 and
w2 within the WordNet graph
If ∄ such P, return 1.0
Calculate the shortest path from either word
to P, d
Calculate the path from P to the root R of
the WordNet graph, rd
Return d / (d+rd)
```

Therefore, the distance is expressed as:

$$sem(w1, w2) = \frac{minDistToCommonParent}{distFromCommonParentToRoot + minDistToCommonParent}$$

The lexical similarity of two words is calculated using the normalized Levenshtein distance between them. The Levenshtein distance between two string s1 and s2, with lengths of $|s1|$ and $|s2|$ respectively, is given by the following equation:

$$lev_{s1,s2}(|s1|, |s2|) = \begin{cases} \max(|s1|, |s2|), & if \ (\min(|s1|, |s2|) = 0) \\ \min \begin{cases} lev_{s1,s2}(|s1| - 1, |s2|) + 1 \\ lev_{s1,s2}(|s1|, |s2| - 1) + 1 \\ lev_{s1,s2}(|s1| - 1, |s2| - 1) \end{cases} \end{cases}$$

We normalize the Levenshtein distance in the [0, 1] space as follows:

$$nlev_{s1,s2}(|s1|, |s2|) = \frac{lev_{s1,s2}(|s1|, |s2|)}{\frac{|s1| + |s2|}{2}}$$

Giving slightly greater importance to the semantic similarity, we assign a weight of 0.75 to the semantic distance and a weight of 0.25 to the lexical distance. Therefore, the overall distance function used by the hierarchical clustering module is:

$$d(w1, w2) = 0.75 \times sem(w1, w2) + 0.25 \times nlev_{w1,w2}(|w1|, |w2|)$$

2.4 SEMANTIC REASONING COMPUTATIONAL TOOLS FOR ASSESSING DIMENSIONS / ASPECTS OF CREATIVITY

2.4.1 CREATIVITY POINTS COMPUTATION

The Creativity Points Computation service is responsible for calculating a score for a text segment, in comparison to other text segments. Thus, its input is a text segment, and a list of text segments. The service returns a double value in the [0, 10] space, calculated as follows:

Let n the number of segments in the input list L. For each segment, we identify the dominant terms and create the graph of dominant terms from the segments belonging to the list. The dominant terms graph is weighted with the distances between these terms. We compute the min-weight closure for each graph, and compare it to the min-weight closure of the input text graph (constructed with the same process). If a min-weight closure of the input text is bigger than the min-weight closure for the graph corresponding to a text in the list, one point is awarded. The sum of the acquired points is then divided by the number n of text segments in the list, and normalized to the [0,10] space.

2.5   SUPPORTING SEMANTIC REASONING COMPUTATIONAL TOOLS

The supporting computational tools do not fall in the aforementioned categories and are not directly used by client applications. Rather, they have been implemented as they are necessary for the providing the functionality of the other Semantic Reasoning Computational tools.

2.5.1   SEARCH ENGINE WRAPPER

The Search Engine Wrapper is used to obtain online information related to a subject defined by the tutor. The learners are then asked to inspect this information and decide which parts of it are useful for their purpose and current state.

Starting from an initial seed, the service uses the Random Word Generator to produce the search query for the underlying search engines. Up to fifty (50) results for this query are returned and their respective URLs are returned from the service.

The Search Engine Wrapper is parameterizable with respect to the search engine that will be used for obtaining the search results. When the SafeContent parameter is set to true, the service invokes dmoz or fragFinn to retrieve web content, while when the parameter is set to false, the Search Engine Wrapper incorporates the Bing Search Engine. To this end, it implements the Bing Search API [7], which allows parameterization with respect to the amount of results returned, the date that the retrieved pages were created, the exclusion of inappropriate or offensive material etc. Further details on the incorporated Safe Search Engines is provided in Section 0, *Techniques for Ensuring Content Safety and Quality*.

*The module is tri-lingual, supporting search queries for **English**, **German** and **Greek** and communicating with the respective versions of the incorporated engines.*

2.5.2   TEXT CLUSTERING

The Text Clustering service uses Hierarchical clustering (as implemented within the Weka framework [8]) to create clusters of these terms that provide indications of the major themes of discussion around the specific topics. Hierarchical clustering is based on the relative distances between the objects that are to be clustered. In our implementation, we used the distance function employed by the Semantic Distance Calculation service.

3.   TECHNIQUES FOR ENSURING CONTENT SAFETY AND QUALITY

As many of the services implementing the C2Learn Semantic Reasoning Computational Tools deliver unregulated Web content, a major concern and requirement for the tools is the aspect of content safety, and particularly for the targeted age groups (K12 education). Hence, the tools should incorporate content quality control mechanisms, for ensuring the suitability and safety of the presented information. This section provides an overview of the mechanisms incorporated in the C²Learn Semantic Reasoning Computational tools that deal with non-moderated content in order to ensure content safety and quality.

3.1   SEARCH ENGINE WRAPPER

As the Search Engine Wrapper delivers URLs of web resources via external search engines, it is an in principle unsafe service. To ensure the delivery of safe web content the service incorporates various measures.

At the first stage, the Search Engine Wrapper examines the value of its safety parameter. If it is set to *false*, the service uses the Bing Search Engine to perform the desired Web search. On the other hand, if the parameter is set to *true*, the Search Engine Wrapper activates children-oriented search engines for searching the web. We examined various Safe Search Engines in terms of their suitability and connectivity, such as AgaKids[2], FragFinn[3], Quintura for Kids[4], KidRex[5], KidsClick[6], etc.

It was determined that we should incorporate a different search engine depending on the desired language, as the examined engines did not provide content in all the languages covered by C^{2}Learn. To this end, the Search Engine Wrapper invokes the DMOZ[7] search when the request refers to content in English or Greek. DMOZ, historically known as the Open Directory Project (ODP) is the largest human-edited directory of the Web. It is constructed and maintained by a global community of volunteer editors. One of the categories edited and maintained by the community is the "Kids and Teens" section, which is the one accessed by the Search Engine Wrapper. When the request refers to content in German, the service invokes the fragFINN[8] search engine. fragFINN started in 2007 as part of the initiative "A Net for Children". It uses a whitelist approach for ensuring that the accessed content is safe and child-friendly. Additionally, the service make use of blacklists and whitelists of web sources (sites) in order to avoid malicious and adult-oriented sites before processing content from them. Finally, the service examine the presence of content labels when visiting a page, and ignore the pages for which the labels indicate the presence of mature or offensive content.

3.2   WEB MINER & CLOUD OF THOUGHTS

The Web Miner and Cloud of Thoughts services deal with textual content, retrieved from the Web of created by users respectively. The services incorporate a blacklist approach for removing offensive / inappropriate terms and not allowing them to be further processed and presented to the end user.

*A list of such terms was constructed for each of the supported languages (**English, German** and **Greek**).*

4.   TECHNIQUES FOR IMPROVING THE PERFORMANCE OF THE SERVICES

All the aforementioned tools were refined in order to optimize their performance and enable their use in real-time responsive applications, like the C2Learn Games using the tools.

More specifically:

– The Thinking Seed Generator incorporate techniques for pruning the search space and traversing the WordNet graph.
– The Web Miner incorporates caching techniques for reducing the delays caused from accessing the web pages returned by the Search Engine Wrapper.

---

[2] http://aga-kids.com/

[3] www.fragfinn.de/kinderliste.html

[4] http://quinturakids.com/

[5] www.kidrex.org/

[6] www.kidsclick.org/

[7] http://www.dmoz.org/

[8] https://www.fragfinn.de/kinderliste.html

– Improvements were made to the parsing features of the Search Engine Wrapper in order to speed up the content cleaning and text extraction functionalities of the tool.

5.   USAGE OF SEMANTIC REASONING TOOLS IN A C²LEARN GAME

This section provides an example on the usage of the C2Learn Semantic Reasoning Computational Tools in the context of a C²Learn game. The example uses Creative Stories, a storytelling game that uses the various C²Learn components.

5.1   OVERALL ARCHITECTURE SERVING C2LEARN GAME SESSIONS

Figure 1 depicts the core components of the architecture along with their interrelations. The components comprising the architecture are the following:

– *C2Space* is the overall environment serving the different distinct C2Learn Games
– The *Game Registry* is part of C2Space and contains and delivers information for the games that are available for usage in game sessions
– The *Game Session Registry* is part of C2Space and stores the information for the game sessions. It is used by the relevant games in order to determine the appropriate enrolment process and setup.
– The *C2Learn Games* are the actual games used by the game session. They are responsible for the enrolment process of the users playing the game either as group members or as single players.
– The *Creativity Profiling Server* communicates with the C2Learn Games and collects the artefacts created by the players, in order to (a) calculate the Computational Creativity Metrics for these artefacts and (b) update the user's creativity profiles.
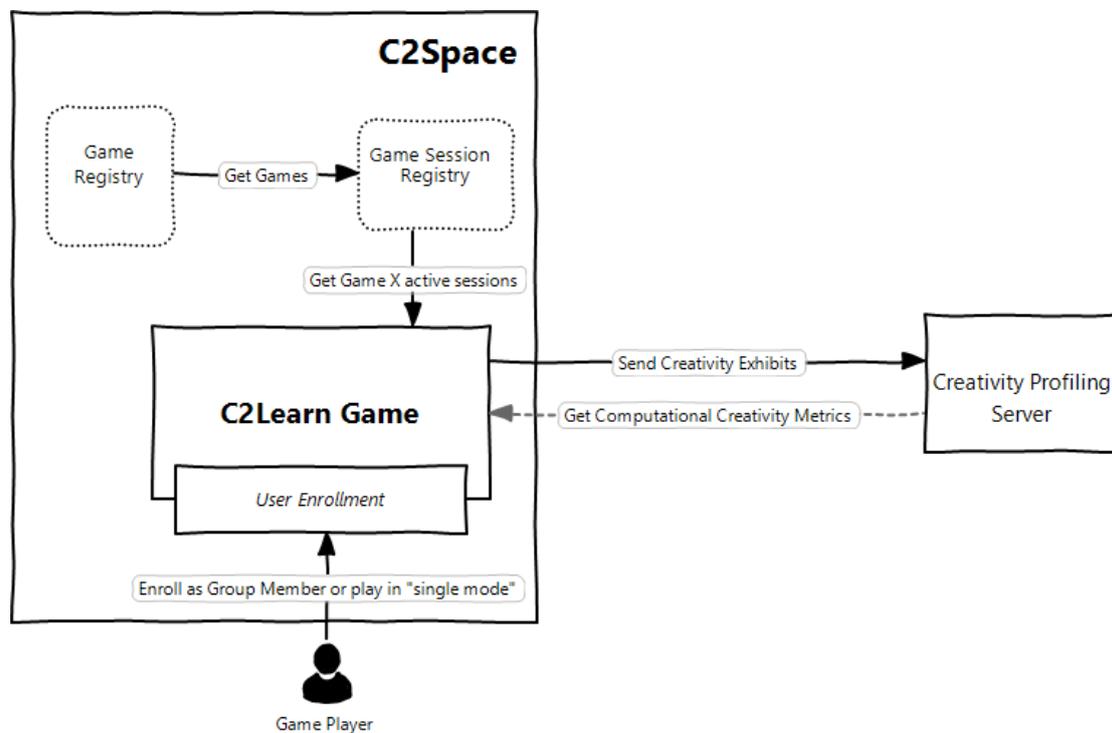
Figure 1: Overall Architecture

5.2   CREATIVE STORIES GAME PLAY

After enrolling in the game session, the players are presented with a multi-panel environment from which they can provide their input, observe the activity of the other groups and get feedback from the computational tools and the teacher. The central panel presents the story fragments created so far by the group (Group 2 in the example) and contains the input field for writing and submitting a new story fragment, along with an indication for the points that will be added for the specific fragment, as they are calculated by the relevant computational tool. In the right-side panel, the players can see the progress of the other teams participating in the game session, along with tag clouds – generated from the Information Summarization Tool – that summarize the stories of the other groups and can be used as inspiration and guidance for progressing with the story.

In the left-side panel, the players can observe their current score and use the input from the computational tools for obtaining input to be used within their story. Figure 2 depicts a mock-up of the game screen in the case that the Creative Input option was selected by the teacher, while Figure 3 depicts the screen in the case that the Competitive Thinking Spaces tool is activated.
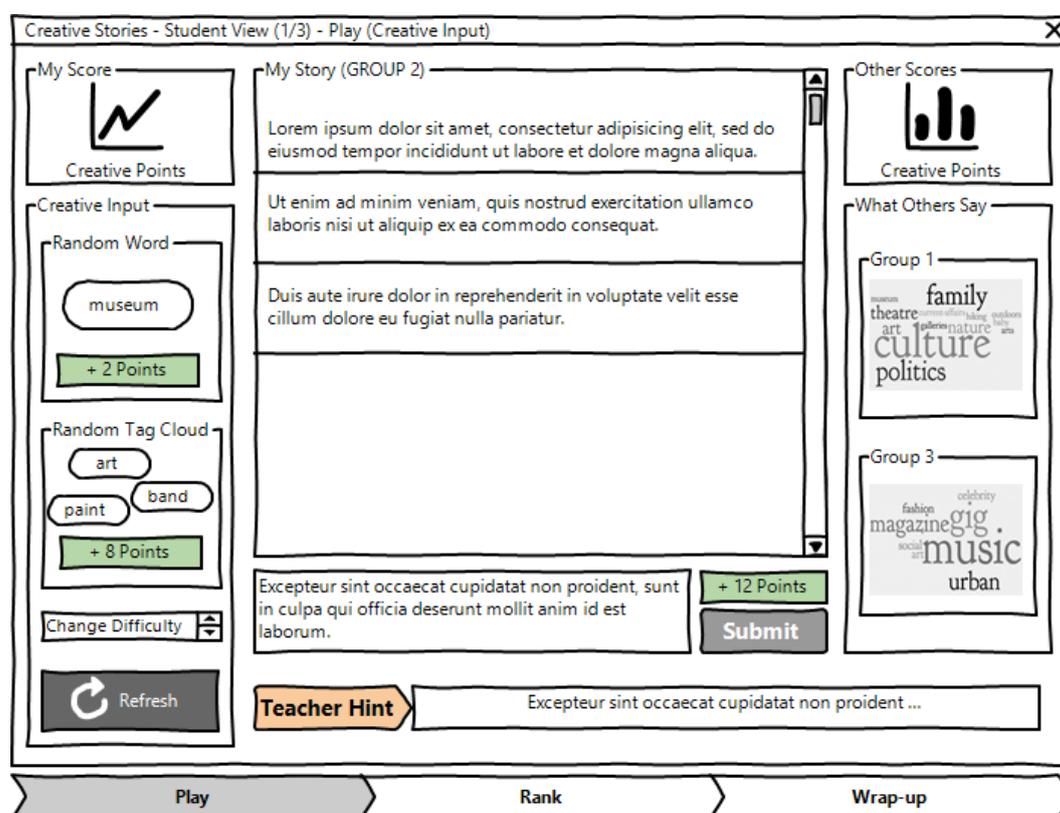
Figure 2: Playing Creative Stories with Creative Input

Figure 3: Playing Creative Stories with Competitive Thinking Spaces

As the groups are playing the game, the teacher can observe their progress and monitor each group's activity (Figure 4). He is presented with the Creative Points Score, as well as the Novelty, Surprise and Impressiveness scores for each story produced by the participating groups, as calculated by the Creativity Profiling Server. Furthermore, the teacher can view details on the usage of the usage of other stimuli from a specific group, and provide a hint to a group for continuing with the story.
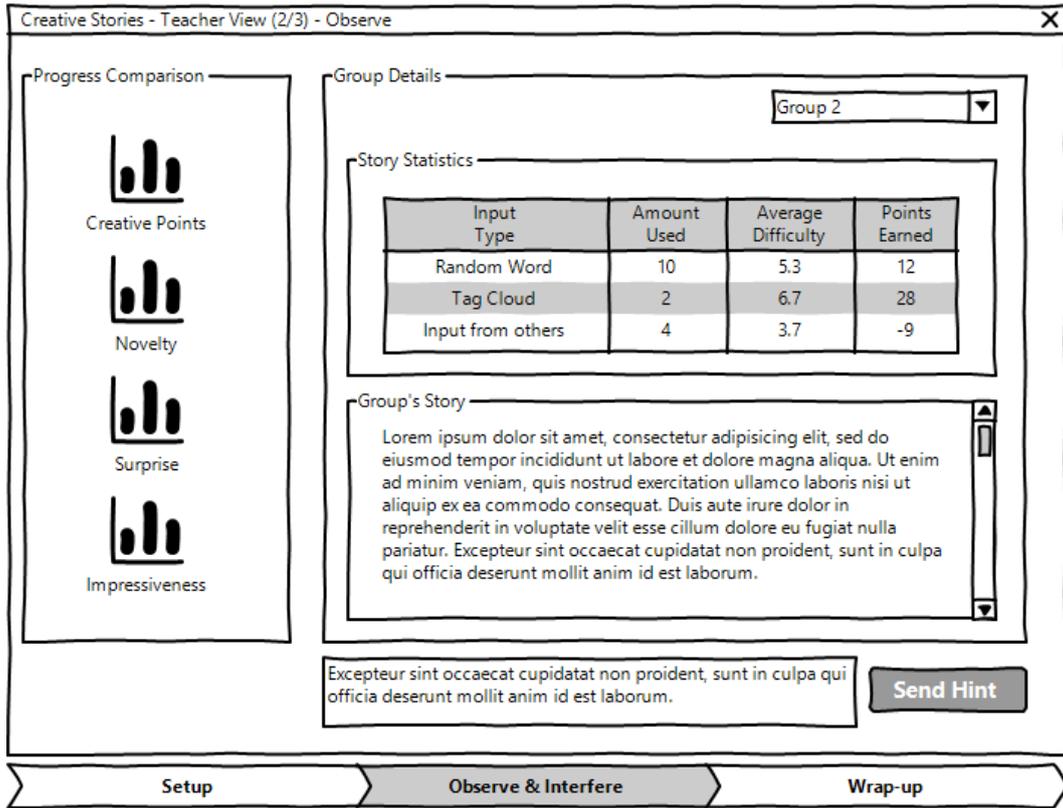
Figure 4: The teacher monitors the groups' progress and provides hints to the players

5.3   CREATIVE STORIES CREATIVE POINTS ASSIGNMENT

During a Creative Stories session, the participating groups are rewarded with Creative Points, determined by their usage of input from the computational tools, as well as, usage of information from the activities of the other players. The Creative Points are defined as the product of the base Creativity Points returned by the Creativity Points Computation service and a modifier that depends on the usage of the aforementioned elements.

We use two distinct functions for calculating the Creative Points in Creative Stories, depending on the type of input selected for the specific Creative Stories session. In the case that the Creative Input option is selected, the Creative Points are given by the following equation:

$$CreativePoints(T) = \left(1 + \prod_{i=1}^{N_T}\left(1 + \frac{difficulty_i}{10}\right) + n + \prod_{i=1}^{N_W}\left(1 + \frac{n * difficulty_i}{10}\right) - 2N_O\right)BasePoints(T)$$

In the equation, N_T is the number of times the player used the Thinking Seeds Generator services, N_W is the number of times the player used the Web Miner services, while N_O denotes the number of words that the player used and appear on the tag cloud created from the other players' stories.

In case the Competitive Thinking Spaces is used as the computational tool input for the game session the equation for the calculation of the assigned Creative Points is the following:

$$CreativePoints(T) = N_{Clusters}BasePoints(T) - \frac{1}{2}N_o$$

where, $N_{Clusters}$ is the number of clusters completed by the specific team, and N_o is the number of words that the team used from the tag clouds summarizing the stories of the other teams.

5.4   CREATIVE STORIES GAME SESSION WRAP-UP

Upon the game's completion, the groups are called to rank the stories of the other teams that participated in the game, by placing the stories in order (see Figure 5). The ranking information is sent to the Creativity Profiling Server, as a creativity exhibit object (for details on creativity exhibits, please refer to deliverable D3.4.x, User Profiling and Behaviour Detection).

Figure 5: Ranking the stories of the other groups

Finally, the game session is wrapped-up and the players and teachers are presented with the results of the gaming session. The teacher then ends the session, with the results of the session sent to the Creativity Profiling Server, which is responsible for computing the relevant Computational Creativity Metrics and updating the profiles of each student based on the performance of his/her group (see deliverable D3.4.x, User Profiling and Behaviour Detection).
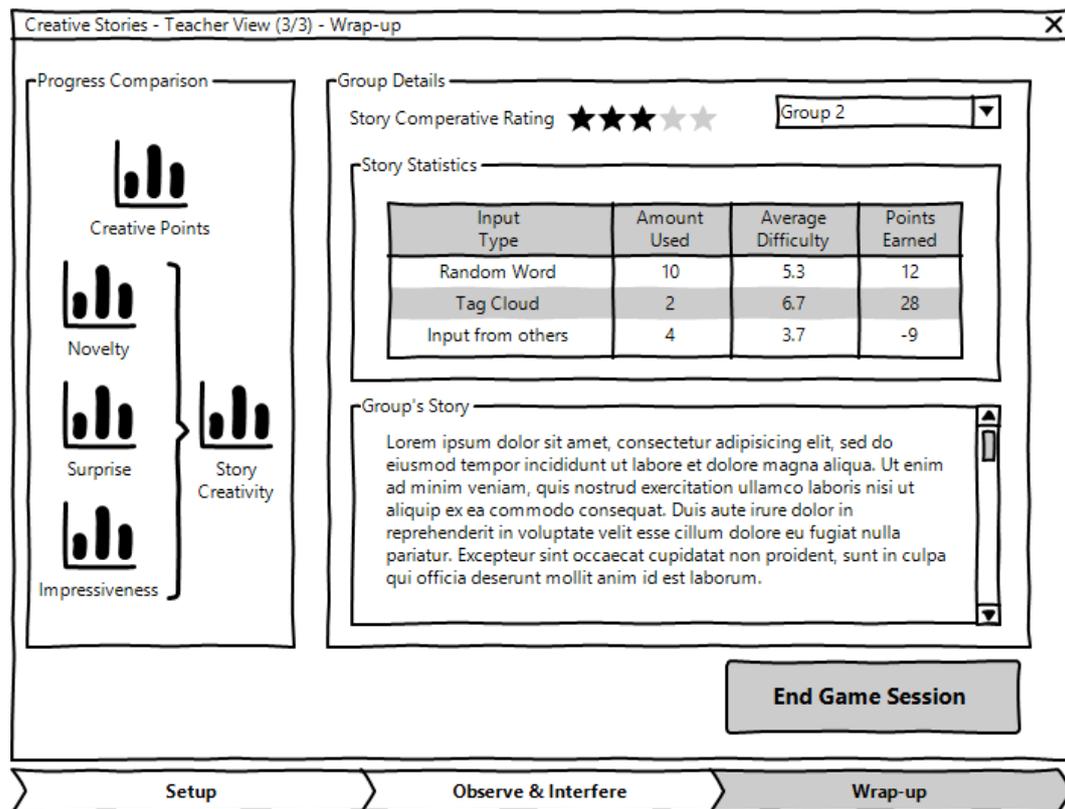
Figure 6: Creative Stories Game Session wrap-up

6.   CONCLUSIONS

The presented C²Learn Semantic Reasoning Computational Tools aim to foster the Semantic Lateral Thinking techniques as defined by the theory, within C2Learn gaming environments.

The tools are categorized as (a) components that provide input for fostering Idea Conception and (b) components for assessing different aspects of creativity. In the first category, the Thinking Seeds Generator, the Web Miner, the Cloud of Thoughts, the Competitive Thinking Spaces, and the Semantic Distance Calculation are exposed as Web Services, able to be called by any client application that .implements the WSDL definition of the services. In the second category the Creativity Points Computation component was implemented and also exposed as a Web Service. Two additional supporting tools, the Search Engine Wrapper and the Text Clustering services were developed to be used by the aforementioned Semantic Reasoning Computational Tools. The tools that handle or produce non-moderated content, e.g. information from the Web, incorporate Content Safety and Quality measures to ensure that the presented content is suitable for the targeted age groups. All the aforementioned tools were refined in order to optimize their performance and enable their use in real-time responsive applications, like the C2Learn Games using the tools.

The usage of the C²Learn Semantic Reasoning Computational tools within different C2Learn Games is defined by the Game Design of each individual game. An example demonstrating the use of the tools in Creative Stories, a storytelling C²Learn game, is described in the present report.

REFERENCES

[1] LATERAL THINKING WORKSHOP – Thinking Tools, http://edwdebono.com/debono/worklt.htm, last accessed July 2013

[2] De Bono E., "PO: A device for Successful Thinking", Simon & Schuster, 1972 and De Bono E., Po: Beyond Yes and No, Penguin Books, 1990.

[3] Gráinne, C. and Fill, K., "A learning design toolkit to create pedagogically effective learning activities", Journal of Interactive Media in Education (Advances in Learning Design. Special Issue, eds. Colin Tattersall, Rob Koper), ISSN:1365-893X, August 2005

[4] Fellbaum, C. (Ed.), "WordNet: An Electronic Lexical Database", Cambridge, MA: MIT Press, 1998

[5] boilerpipe – Boilerplate Removal and Full text Extraction from HTML pages, https://code.google.com/p/boilerpipe/, last accessed July 2013

[6] Snowball, http://snowball.tartarus.org/, last accessed July 2013

[7] Bing Search API–Windows Azure Marketplace, http://datamarket.azure.com/dataset/bing/search, last accessed July 2013

[8] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., "The WEKA Data Mining Software: An Update", SIGKDD Explorations, Volume 11, Issue 1, 2009