Abstract—Traditional mind map uses ordered hierarchical structure to obtain “clarity of thinking”, but it lacks of the subtle quality of the human’s thinking process such as association, comparison, summarization and reasoning which is vital for innovational activities. To address this issue, a mesh model framework of "the next generation mind map" (Mindmap-NG) is thus proposed to provide better help for effective knowledge management and effective thinking. Nodes in mindmap-NG are constructed initially as a tree structure but with additional attributes for back-linking to the other nodes. These back links can then be weighed by means of accessing frequency and content relevancy etc. When a node is visualized into focus, the hierarchical neighbors as well as the mesh links are shown. Additional operations such as searching and grouping can be performed on nodes' attributes, enabling brain-like activities such as association, back-tracing, comparison and so on. A prototype test verifies part of its functionality, yet its cognitive significance needs further inspection.

Keywords-argumentation tool; mind map; mindmap-NG; innovation methodology; lateral thinking; computer-supported creativity

I. INTRODUCTION

A mindmap (or mind map) is a tree-like diagram used to represent words, ideas, tasks or other items and their relationships [1]. Mindmaps are used to generate, visualize, structure, and classify ideas, and as an aid in study, organization, problem solving, decision making, and writing. Fig. 1 is an example of a hand-draw mindmap with 5 top-level branches.

Figure 1. A hand-draw mind map

Figure 2. A mindmap created by Buzan's iMindMap

Compared with the hand-draw versions, computer-generated mindmaps are more subtle and versatile. There are over half a dozen free mindmap software, among which the opensourced FreeMind software is a typical one [4]. It offers an elegant yet effective way for managing meeting plans, note taking, problem solving, etc.

With the help of computer software, one can produce complicated yet graceful mindmaps. Fig. 2 is a mindmap of someone's recreation plan, produced by iMindMap [3], a proprietary software created by Tony Buzan, who claimed to be the founder of the modern mind mapping.

Despite the variety of styles of visual manifestations, modern mindmap remains its tree-like structure and isolates itself from modern technologies such as DBMS and data mining. Few of them can deal smoothly with complicated systems of more than 500 objects [2], or have "database connectivity" or "numerical analysis" ability. Even the most expensive mind mapping software is void of the lateral analysis capabilities that excavate the nexus between different branches; yet these capabilities are of great help in creative activities such as brain storming and "lateral thinking" (the latter of which is an innovation methodology created by De Bono [7]).

The rest of this paper is organized as follows. Section 2 describes the application areas of mind mapping and their urgent demands. Section 3 is the data structure proposed for the new mindmap-NG framework. Section 4 discusses operations that can be performed on mindmap-NG. Section 5 is a test implementation and Section 6 is the conclusion and future work.
II. APPLICATIONS OF MIND MAPPING

Mind maps have been widely used in both academic and business fields for decades, especially in areas where creative thinking & innovation is an important issue. Fig. 3 shows part of the result of the Mind Mapping Software Survey [5]; it is clear that the majority of people use mindmap software for project-related applications such as planning, presentation, note taking, problem solving, decision making, etc. Another big application area is education/researching, where knowledge management and innovation is of dominant importance.

Another survey concerns about the web-based mind mapping software [6]. Although still in its primitive stage - web-based mindmapping software failed the expectations of even the most enthusiastic user before 2007 - but the survey also points out the most promising direction to which the future mindmappings should be developed. It shows that the most compelling requirement about mind mapping software is to enable collaboration, that is, concurrent updates to a single map over the Internet. Collaboration is vital for innovation activities such as brainstorming, and is a must for education. Experiments have shown that utilizing web-based mindmapping for collaborative learning and researching in university could have a "booming efficiency" [8]; some researchers even built special application to support collaborative mind mapping for creative activities [9].

The second most important requirement is to allow multiple formats of data to be associated with one node, including images, sounds, documentations, hyperlinks, files and numerical data. The "on-spot" presentation of these data is also required during a business or innovation process.

More similarity to human thinking process is also required. In fact, some users have already suggested that a mindmap software should be "intuitive, fit with the way I think and represent things and their relationships" [5].

With various kinds of data included in nodes, corresponding data processing procedures could be performed, such as locating, sorting, grouping, etc.; these procedures can then form the basis of application level operations such as searching, comparing, filtered visualization & reporting, as well as other innovational activities.

III. DATA STRUCTURE OF MINDMAP-NG

Traditional mindmap is in essence a tree structure of data nodes. The number of nodes is often limited, so we can use a simplified model to represent this structure, as shown in Table. 1.

Each node has a unique id to identify itself, and a parentid to indicate its hierarchy level. The content field contains (mostly) text data; this is also the visible text shown in the mindmap. VisAttr determines the node's display format such as location, color, font, icon etc.

Mindmap-NG needs additional fields to describe relationships between lateral nodes and operations that can be performed on them, see Table. 2.

Tags is a blog-like text string to describe the nature of the specified node; e.g., a tag like "notes; data mining; outlier" indicates that this node is a "note" about data mining technology and outlier detection. When a "main branch" node is created, the "tags" attribute must be explicitly assigned. When a child node is created, it inherits the tag of its parent; the user can then optionally edit this tag.

Routeinfo is a simple text description about the route from the parent node to the child node; this description helps the user to understand the reasoning process of the mindmap.

Content is now a complex field consisting of multiple types of data, and a file system of the data files.

Links is the most import field in the mindmap-NG framework; it is a compound of single links; each link consists of the following fields:

- Peerid: the peer node to which it links;
- Weight: the weight of the route to the peer;
- Description: a descriptive text about the route to the peer.

Table I. Data Structure of Mindmap Nodes (Simplified)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Unique id for a single node</td>
</tr>
<tr>
<td>parentid</td>
<td>Pointer to parent node</td>
</tr>
<tr>
<td>content</td>
<td>Text data</td>
</tr>
<tr>
<td>visAttr</td>
<td>Visual attributes</td>
</tr>
</tbody>
</table>

Table II. Data Structure of Mindmap-NG Nodes

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Unique id for a single node</td>
</tr>
<tr>
<td>parentid</td>
<td>Pointer to parent node</td>
</tr>
<tr>
<td>tags</td>
<td>Blog-like tags; inheritable</td>
</tr>
<tr>
<td>routeinfo</td>
<td>Information about route from parent</td>
</tr>
<tr>
<td>content</td>
<td>Text content &amp; related data, including: sound, image, documentation, files, etc.</td>
</tr>
<tr>
<td>links</td>
<td>Peerid</td>
</tr>
<tr>
<td></td>
<td>Weight</td>
</tr>
<tr>
<td></td>
<td>Description</td>
</tr>
<tr>
<td></td>
<td>Peerid</td>
</tr>
<tr>
<td></td>
<td>Weight</td>
</tr>
<tr>
<td></td>
<td>Description</td>
</tr>
<tr>
<td>visAttr</td>
<td>Visual attributes</td>
</tr>
</tbody>
</table>

Links are dynamically created when a corresponding operation is performed; for example, the "find similar nodes" operation will create a link from the current node to each
similar node found by this operation; the "weight" factor of each link is also increased as a result of this operation.

*Weight* is a measure of significance of the corresponding link; it is determined by the number of visits to this link. The bigger the weight factor is, the more likely that the link is an important pattern.

IV. OPERATIONS OF MINDMAP-NG

Under the above proposed mindmap-NG framework, the following operations may be performed on the node structure mentioned in Section 3:

A. Searching

Searching is the basic operation of all other operations; it can be performed mostly on *tags*, yet it can also be performed on the text component of the *content* field. When a search result is formed, a new top level node is created (if not already exists) with its *links* pointing to all found nodes. The visualization subsystem then displays the search result with all found node as the children of the newly created "search result" node.

Fig. 4 illustrates the search result for the word “outlier” on the “content” field of a simple map.

B. Grouping

Grouping is a variant of “searching”; it groups nodes by one or two of its tags. As we mentioned before, the blog-like tags specify the nature of the corresponding node and can be used as a classifier; grouping on nodes in different branches (or nodes belong to different users in a collaborative environment) could be very useful for collaborative creativity.

C. Find Similar Nodes

We define the *similar nodes* to be those nodes whose tags (key words) are identical. This operation is a variant of grouping with the whole *tags* string being used as the grouping criteria.

D. Find Sparse/Intensive Path

Sparse paths are those with low weight factors; intensive paths are those with high weight factors. Although sparse path hardly has any meaning in logical deduction, but it plays an important part in brainstorming and lateral thinking. Intensive path, on the other hand, indicates a frequently accessed link, and provides useful information for association rule analysis.

E. Association Analysis

This operation is based on the result of the above “find sparse/intensive path” operation. An intensive path implies strong logical connection between the parent and the child. With additional grouping criteria (such as “similar nodes”), intensive paths can likely be the association rules to be sought.

V. TEST IMPLEMENTATION

As a simple prototype, we just used pre-defined text values to test the proposed model. Using the Java programming language, nodes are constructed by a class *MMNGNodes*; its basic structure is shown in Fig. 5.

The complex field of node’s *content* is stored as an object of class *MMNGContent*; aside from the ordinary String data such as content text and *hyperlink* text, other data are stored as files with their path in one vector and the corresponding data types in another vector. Methods are implemented to traverse these data objects.

*VisAttr* is implemented as a Hashtable; font name, font color, node’s location on canvas etc are stored as key-value pair.

*Links* is implemented as a Vector; for each link created, an object of type *MMNGLink* is stored into this Vector.

All nodes are stored in a vector; we use pre-defined values to perform 2 types of experiments: one is the “search” operation and the other is to find which link has the biggest weight. We have to do a full traverse for the second test; this hints that our proposed class implementation needs to be further refined.
From this primitive experiment, we can hardly draw any
cognitive conclusion, only to verify some of its technical
functionality; more intensive experiments in a social level
must be conducted in order to justify this kind of computer
supported creativity.

VI. CONCLUSION AND FUTURE WORK

Traditional mind map is in essence a tree structure whose
nodes are virtually text objects with visualization attributes;
the nodes are arranged in hierarchical manner to help
establishing a clear representation of the managed content,
but it lacks the ability of lateral analysis for nodes in
different branches. To address this issue, a new mindmap-
NG framework is proposed and its functionalities are
discussed. This framework is tested on a limited level, but its
usefulness in lateral searching, grouping and comparison etc
holds. These lateral analysis ability can be of great
significance for innovation and creativeness.

Although our approach is still primitive, it adds some
crucial functionality that is needed for effective & creative
thinking to the already prosperous mind mapping software
family. Future work includes the following:

1) Complete the user interface & visualization
subsystem.

2) Enable database connectivity for large number of
data objects and multiply users.

3) Enable collaboration via Internet.

4) Implement the lateral thinking methods proposed in
[7].

5) Implement the eight cognitive maps proposed in
Thinking Maps® [10].

REFERENCES

Software,” in Systems Engineering and New Developments in
[8] Dmitrij Koznov, Michel Pliskin, "Computer-Supported Collaborative
Learning with Mind-Maps", in: Leveraging Applications of Formal
Methods, Verification and Validation, Third International Symposium,
88479-8_34.
[9] Stéphanie Buisine, Guillaume Besacier, Marianne Najm, Améziane
Aoussat and Frédéric Vernier, "Computer-Supported Creativity: E-
valuation of a Tabletop Mind-Map Application", in Engineering
doi: 10.1007/978-3-540-73331-7_3.
[10] David Hyerle, Thinking Maps®: A Visual Language for Learning, in:
Knowledge Cartography, Springer, 2008, doi: 10.1007/978-1-84800-
149-7_4.