

# Seminarwork Mind Maps

Christoph Rissner

March 3, 2003

## Abstract

Mind Maps as introduced by Tony and Barry Buzan are hierarchically organized maps giving an overview of large data collections. Originally used in learning and memorizing applications Mind Maps are nowadays used for task planning, organising, presentation application and lots more. After a brief introduction to Mind Mapping this paper will concentrate on computer based Mind Mapping applications.

After an short overview I will take a closer look on two applications. As a very popular application, Mindjet MindManager is chosen. Being a commercial application this one comes with rich featureset and after a general introduction I will take a closer look at the MMScript scripting capabilities. The second application I will introduce is FreeMind, an opensource approach written in java. Although providing less graphical markup features FreeMind comes in handy where the principles of mindmaps are required and additionally FreeMind uses a cleartext XML format for its data files.

The next part leads to data interchange between these applications. The MMScript scripting capabilities of MindManager allow to access the MindManager application as well as its documents through a COM interface. Utilizing the so called "MindManager Open Interface" almost any functionality the graphical user interface provides can be executed programmatically. This is the starting point for the import and export scripts. Additionally a suitable XML Schema will be developed. The last step for data exchange between MindManager and FreeMind consists of two XSL transformations capable of converting a file exported from MindManager into a file that can be read by FreeMind and vice versa.

The last part takes a look at alternative visualization tools. As a main step I will concentrate on transformations into docbook format. This is not only another intermediate step but a point from where on one can make use of established transformation tools to finally reach popular electronic publishing formats like HTML, PDF and as a new concept XSL Formatting Objects.

# Table of Contents

<b>1</b>	<b>Concept</b>	<b>5</b>
1.1	Definition . . . . .	5
1.2	What . . . . .	5
1.2.1	What is a Mind Map? . . . . .	5
1.2.2	How to Mind Map . . . . .	6
1.2.3	Examples . . . . .	6
1.3	Who . . . . .	7
1.3.1	Tony Buzan . . . . .	8
1.3.2	Barry Buzan . . . . .	8
1.4	Why . . . . .	9
1.4.1	The human brain and Mind Maps . . . . .	9
1.4.2	Structurized Mind Mapping . . . . .	10
<b>2</b>	<b>Applications</b>	<b>11</b>
2.1	Computer based applications . . . . .	11
<b>3</b>	<b>Software</b>	<b>12</b>
3.1	Mindjet MindManager . . . . .	12
3.2	FreeMind . . . . .	13
3.3	Other applications . . . . .	13
3.3.1	Mind Map Plus . . . . .	13
3.3.2	Inspiration . . . . .	14
3.3.3	Visual Mind . . . . .	14
3.3.4	SimTech Systems MindMapper . . . . .	14
3.3.5	Eminec MYmap . . . . .	14
3.3.6	ConceptDraw MINDMAP . . . . .	14
3.3.7	MyMap for *nix . . . . .	15
<b>4</b>	<b>My Work</b>	<b>15</b>
4.1	The VB Scripts . . . . .	15
4.1.1	Exporting MindManager data to XML . . . . .	16
4.1.2	Importing XML data to MindManager . . . . .	17
4.1.3	A Mind Map XML Schema . . . . .	18
4.1.4	Data Exchange . . . . .	19

4.2	Passive Visualization . . . . .	20
4.2.1	DocBook . . . . .	20
4.3	What remains to do . . . . .	22

# 1 Concept

Mind Maps as introduced by Tony and Barry Buzan [MindMap] are hierarchically organized maps giving an overview of large data collections. Originally used in learning and memorizing applications Mind Maps are nowadays used for task planning, organising, presentation application and lots more. After a brief introduction to Mind Mapping this paper will concentrate on computer based Mind Mapping applications.

## 1.1 Definition

A Mind Map is an expression of radial thinking and therefore a natural function of the human mind. It is a graphical method. A Mind Map has four basic attributes:

1. The object of attention is captured in a central image.
2. The objects main topics radiate away from the center like branches.
3. Branches contain key images or words that are written on a line connected to the central image. Subtopics are written on deeper level branches. These branches are again connected to the higher level branches containing higher level topics.
4. Altogether the branches build a network of interconnected nodes.

## 1.2 What

To get a better understanding of Mind Maps I start with a description of what I am talking about in this paper and give the reader an impression to the question:

### 1.2.1 What is a Mind Map?

Basically a Mind Map is nothing more than yet another style of taking notes. The user simply starts with his topic of interest and than recurses down into the different aspects regarding this matter. The difference to traditional techniques is not found in the content, it is solely in the representation.

The main matter is found in the center, multiply colorized and therefore immediately attracting the viewers attention. The subtopics radiate away from the center, building a tree like structure where any child branch radiating out is associated with a specific subtopic. As we climb down this tree topics get more and more specific therefore building a strict hierarchy of semantics. The creator of the Mind Map is encouraged to richly use colors, symbols, drawings, personal codes and any type of markup that helps him to describe this topic.

What for now simply looks like a childish way of note taking, a way for people that are not able to order their thoughts turns out to be an almost revolutionary way of organizing ones thoughts by simply not writing them in traditionally structured, what mostly means linear, but as a beautiful colored image.

### 1.2.2 How to Mind Map

- Start with a blank sheet of paper, size approximately DIN A3. One should not feel restricted because of too less free space.
- Start with the main matter in the center. This one should be a descriptive image with at least 3 colors.
- Use images when possible, otherwise use only keywords.
- Subtopics are placed on lines radiating away from the map center so that they create new centers of this special subtopic. Put the keywords on the lines to enforce the structure of the map.
- Use print letters rather than script, use lower case letters to get an improved visual distinction. Also notes can then be better remembered.
- Think three dimensional and borderless. Paste additional paper when there is more space needed. Do not think in paper size dimensions.
- Use colors to depict themes. Use colors to depict associations. Use colors to make things stand out. Simply use colors.
- Anything that stands out visually on the map will stand out in your mind.
- Use icons, codes and other visual aids to establish links between unrelated elements.
- Add ideas as they occur wherever they may fit in. Neither hold back ideas nor judge them.
- To not get stuck in one topic assure to switch to other topics early.

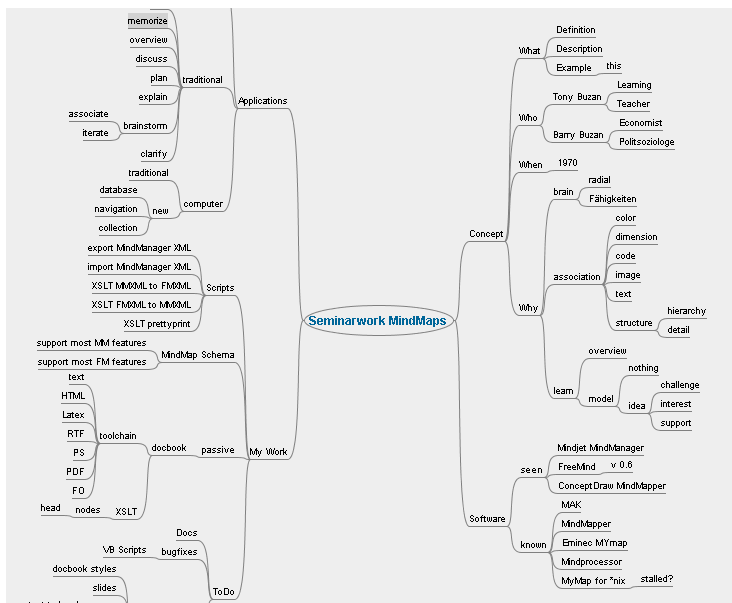
### 1.2.3 Examples

The following to Mind Maps intend to give the reader an impression of what I am talking about. One Mind Map is a screenshot from FreeMind, a Mind Mapping computer application showing the structure of this document. The second one from Tony and Barry Buzan giving an impression of the structure of the Mind Map book. As you may guess this one is done by hand.

#### The Mind Map Book structure



Mind Map of this document



### 1.3 Who

Although it does not really address any subject covered in this paper, it may be interesting to know who first came along with Mind Maps. Besides one can see the first main applications of Mind Maps because they were not developed coming from an academic point of view but from simple problems anyone knows and can understand.

### 1.3.1 Tony Buzan

Tony Buzan was born in 1942 in London and is the inventor of Mind Maps. In 1964 he graduated at the University of British Columbia in Psychology, Anglistics, Mathematics and Allgemeine Naturwissenschaften.

Sometime during his second year of study he recognized that the overwhelming amount of material simply overloaded his brains skills. While others did, and as we all know still try to, accept the fact and try to make the best of the situation, Tony Buzan informed himself about existing learning techniques and especially asked for literature about effective brain usage. It sounds quite funny but he was directed to the medical department for further information.

At this point he recognized that he had entered a completely new field and decided to learn about the learning process itself, techniques of memorizing, creative thinking and the general nature of thinking.

During his research including fields like psychology, semantics and different sciences regarding the human brain in all its aspects he found that the brain works better when using its physical and intellectual skills in combination. The first consequence for his own learning style was to simply use colors in his notes and instantly he found that the memorizing effect increased by more than 100 percent.

The further consequence was that not only that the memorizing effect increased but he began to feel joy about learning again. These facts were motivation enough to further investigate and possibly find the ultimate technique of learning and thinking.

### 1.3.2 Barry Buzan

Tony Buzan's brother Barry first learned about the concept of Mind Maps in 1970. While Tony Buzan used them as a memorizing helper Barry was more interested in the aspect of Mind Maps for creative thinking and categorizing his own thoughts while writing on his doctoral theses. He soon figured out that the gap between thinking and actually writing was one of the main problems and a critical point on his way to success.

He recognized that using this new technique gave him the capability to separate the whole process of thinking and writing. He could organize his thoughts with more ease and give them structure without the need to reorganize his documents. He was able to think about his work more clearly and more comprehensive.

He then compared his way of writing to his colleagues and immediately saw the advantage. While they used traditional approaches where writing and rewriting of there work used up their time and their creativity suffered under the heavy workload they generated themselves, Barry Buzan not only had a very clear concept when he started actually writing his thesis. He had already worked out the complete structure and at any point clearly knew the direction he wanted to elaborate.

It is quite nice to hear that his very successful way of composing his doctoral thesis lead its completion in less than the maximum time of 3 years. So he found



time to co-publish a student paper, create a new paper about international relationships, learn to motorcycle and last but not least to marry.

Barry Buzan states that since then Mind Mapping is a central part of his scientific work. He sees himself able to write clear articles covering abstract topics which complex nature often leads to very confusing style.

## 1.4 Why

### 1.4.1 The human brain and Mind Maps

The human brain has several intellectual skills which are distributed over the cerebral cortex. And despite this fact some skills dominate on either of the left and the right half. The right half dominates for the creative skills like color, rhythm, dimensional thinking and imagination. The left half goes for the logical skills like numbers, words, lists and sequences.

It turned out that during a learning process the brain can mostly remember the following things:

- Primacy effect  
Things learned at the beginning of the learning process.
- Recency effect  
Things learned at the end of the learning process.
- Associations  
Things that can be associated with patterns or already known things. It also helps if these things can be associated to other data that has to be learned.
- Markups  
Uniquely marked or stressed things that stand out its environment.
- Things that directly address on of the five senses.
- Things that high personal interest.

Another point is that the human brain tends to automatically complete things it encounters. Whether these are unarticulate words it can understand anyway or read undistinguishable letters the brain can recover lots of information based on already known facts. This can be done through a very high level of associations.

Mind Maps are designed to attract the viewers attention by more than its content. First of all one should rather use images or symbols to point out a semantic. Early studies published 1970 from Haber show that the average recognition rate of previously shown images lies between 85% and 95%. Therefore 2560 images were shown for 10 seconds. Later the experiment was repeated where the images were shown for 1 second but the results did not change.

Further studies by Nickerson showed that the brain is almost perfectly capable to recognise images. Experiments were taken with 10000 different images shown

to test persons. After that any proband was confronted with pairs of similar images where only one image was shown previously. The recognition rate was 99.9%. Further estimations say that tests with 1 Million images would lead to a recognition rate of 98.6%. It is important to mention that the images used have a *living* character. These images could be described as *outstanding* and *memorable*. These are highly subjective attributes and therefore cannot be defined generally.

The reason for that is that images use a broad field of cortical skills. Images consist of shapes, these shapes commonly render threedimensional objects. And after all, the real world simply is threedimensional (although different opinions exist). Finally objects are defined by additional attributes like their color and their character. But images use even more of our skills. They use *visual rythm* and, whats most important, our imagination. All these facts together are more powerful to trigger an association chain than words.

Of course images are often an inappropriate way to express some special interest. May it be the abstractness of a topic or one can simply not associate an image to the current topic. Of course written words are an acceptable way to work around. One should only pay attention to not write whole sentences, neither partial sentences are good. The really important information can be fully associated from a single keyword.

One is encouraged to use any other aid to clearly express a subject as long as it does not cut the creative thinking too much.

#### 1.4.2 Structurized Mind Mapping

Now that we have identified the elements of a Mind Map we come to the part where we want to combine them to finally build the Map. Therefore we have to categorize and struturize the available information.

A study by Bower, Clark, Lesgold and Wimzenz taken in 1969 proves the importance of structure as a memorizing aid. 28 different words were presented to two groups of test person. While one group got the words in random order the other group saw the words structured hierarchically by going from the most general word at the top down to specific incarnations of this topic. Afterwards the group confronted with the structured list was better able to remember the words.

It is therefore an important aspect for Mind Mapping that the whole information is categorized and structured. Mind Maps effectively use this fact in the way how the elements of a Mind Map are connected. Any main topic regarding the Mind Maps subject is connected to the central image with a line. Subtopics in turn are connected to their respective higher level topics with lines. Using this simple procedure results in two effects. First, any topic treated in a Mind Map is categorized thematically and has an implicit level of detail through its depth in the whole graph. The second effect is that while a Mind Map gives a rough overview over a subject through its first and second level topics a very detailed view can be given through its deeper levels.

## 2 Applications

The range of Mind Mapping applications almost seems infinite. Since their claim is to replace (and enhance) someones personal style of taking notes or, maybe to be more precise, creating documents generally they encourage the user to use them anywhere. Of course there exist typical applications including:

- taking notes
- organizing ideas, the own ones or other peoples
- evaluate different choices and help to make a decision
- memorize things, e.g. learning or teaching
- explain ones point of view
- discuss topics
- give overviews and presentations
- story-telling, self analyzing, ...

The fact that Mind Maps provide overviews at the same time as very detailed information makes them especially usable for giving brief overviews while immediately enabling detailed discussions.

Another point not mentioned above is, probably the most popular application for Mind Maps, is Brainstorming. There exist numerous techniques that give guidelines of what to do while brainstorming, how to start, when to stop and what to do with the results.

Mind Mapping has the direct access to brainstorming since no further guidelines or tools are required. Mind Maps utilize the humans creativity by means of the brains association skills.

The Mind Mapper simply adds his thoughts word by word to its brainstorming map and repeatedly adds all the associations that come to his mind as child branches. The generated map is structured without any further effort because all the associations are already in their right place.

This sounds very simple but of course its necessary to do several iterations to get to the aimed target.

### 2.1 Computer based applications

Coming from traditional applications it is clear that not only computer industry found a market for Mind Mapping computer applications, different approaches similar to Mind Mapping came up independently, starting from commonly known visualization tools like tree views and different incarnations of map views. Most of these approaches however don't target at the typical Mind Mapping applications but rather on visualization of data collections.

As usual in the evolution of computer based applications the first implementations are not much more than a port from real to binary life.

The second edition of the Mind Map Book was released in 1997. It comes with a few screenshots showing an application developed in cooperation with the Buzan brothers called *Mind Map Plus*. Even Tony Buzan has to admit that the visual possibilities and the ease of usage could not compete with handdrawn Mind Maps.

But then there are advantages that gave enough motivation to further investigate the possibilities in computer Mind Mapping. These obviously include the possibility to automate the Map creation process as well as to edit the Mind Map. It further enables the user to completely restructure the Mind Map and try several variations with almost no effort.

The next chapter gives a brief overview of existing software packages.

## 3 Software

Several applications exist specifically designed to bring the approach of Mind Mapping to a computer's desktop. They all share the strict concept of Mind Mapping although having different goals. As a part of my seminar work I have taken a look at some applications and chose two specifically to further elaborate my work.

### 3.1 Mindjet MindManager

This is a commercial application running on Microsoft Win32 platforms. It is an easy to use WYSIWYG application and aims towards project planning, resource planning and related fields. It seems like this one currently is the most popular Mind Mapping application available and at the same time the most comprehensive application too.

The obviously targeted user group are people in company management positions since it comes with a whole bunch of tools for project planning and organizing tasks as well as simple but yet quite powerful tools for creating presentations and exporting a Map for further use into the Microsoft Office Toolchain.

Main features of the Mindjet MindManager are:

- Markups  
MindManager comes bundled with lots of different icons, marks, symbolic codes and a complete set of symbol images one can utilize.
- Map creation  
MindManager supports the user through templates, wizards and a good help system.
- Map modification  
MindManager has a new filter and select system suitable for map data handling.

- Interoperability  
Integrated support for the Microsoft Office toolchain.
- Presentation  
MindManager features a presentation mode as well as export facilities to Microsoft Powerpoint or HTML based presentations.
- Scripting  
Support for customization and extensions through the OpenInterface, a Windows COM interface for automatic task handling. Additionally MindManager comes with a Visual Basic Script interpreter to write and run such scripts.

The last point mentioned, namely the scripting capabilities, is the starting point for my work and I have written two scripts that allow export to a non proprietary file format discussed later on as well as import of these files to MindManager therefore allowing better data exchange between MindManager and applications not supporting Microsoft file formats.

## 3.2 FreeMind

This is a Java application originally started by Jrg Mller and is now maintained by Daniel Polansky. FreeMind is an open source project released under the GPL and evolves to a general editor and viewer for general tree structured data.

Jrg Mllers original vision states that freemind shall become for tree structured data what emacs is nowadays for linear data (i.e. text). Daniel Polansky admits that this is a rather far fetched vision for that it would require to provide scripting capabilities and at least basic operations like upcase, downcase, replace and many others. He even states that this goal might not be of great value because lots of basic operations exist that might have nothing to do with the idea of FreeMind.

For my seminar work I use FreeMind as the second active Mind Mapping tool. I will try to setup a form of data exchange between FreeMind and Mindjet MindManager.

## 3.3 Other applications

This section gives a short overview of applications I have come across but either was not able to evaluate or seemed slightly misleading regarding my seminar work but are still worth to be mentioned

### 3.3.1 Mind Map Plus

Very view information can be found on this one. But because of the fact that it is mentioned in the Mind Map Book I assume it was one of first ones available. It features user guidance in the Map creation process and allows Map modifications and links to external text files as well as data export to text files.

A very nice feature is a special menu for group Mind Mapping. This consists of a menu using large fonts and enables a whole group to follow screen actions.

### **3.3.2 Inspiration**

This commercial application features three edit modes (diagramm, text outline and text note mode) which are similar to MS Powerpoint edit modes. It comes with a basic symbol library and supports the tagging of nodes with action codes. Furthermore it supports text notes and a few different types of links attached to a node. The most interesting feature for Inspiration seems to be the ability to attach sounds to a note. There is support to record and playback sounds for each node. Export functions to some image and rich text formats as well as to HTML presentations make Inspiration a complete solution.

### **3.3.3 Visual Mind**

Visual Mind is a commercial application which user interface strongly remembers on the Mindjet MindManager. It includes the basic functions for Map creation and modification and the possibility to attach text notes. Visual Mind offers export into several formats including HTML and XML. There even is an export option to a selfextracting ZIP file that contains a Java applet resembling a file viewer for Visual Mind documents. In fact it seems like it was an earlier version of MindManager.

### **3.3.4 SimTech Systems MindMapper**

This is a Win32 application similar to the Mindjet MindManager. It also features the integration into the Microsoft office toolchain and even comes with more builtin data import/export functionality, as a highlight I'd like to mention the export and import facilities using a XML format. Unfortunately in the demo of the current version 3.2 seems a bit unprofessional. It lacks elementary parts as the help and declines to open its own tutorial maps.

### **3.3.5 Eminec MYmap**

This is another Microsoft Win32 application that sees itself as a supplementary for the Microsoft Office toolchain. It positions itself for at the top of the toolchain just to support brainstorming and organizing tasks that should be fed into Microsoft Project and then into the subsequent applications.

Since MYmap could not be run after installation I was not able to investigate any further.

### **3.3.6 ConceptDraw MINDMAP**

This one is very similar to Mindjet MindManager as well as its layout but also in its features. In addition to all the Mind Mapping specific features it provides vector drawing functionality which lets the user build a complete presentation.

### 3.3.7 MyMap for \*nix

This aims to be a mindmapping program for \*nix This is some project started by Jost Schenk apparently sometime before Jan. 2000 and seems to have stalled development.

Unfortunately it still uses Qt 2.2 and KDE 2 so i couldn't get it to compile neither on my PC nor on pluto.

## 4 My Work

The goal of my seminar work was to bring some interoperability between active Mind Mapping applications as well as to utilize passive visualization tools such as Latex.

I started my work by exploring the MindManagers OpenInterface. This is a COM interface that allows to programmatically use MindManager functionality in almost any aspect a user is able to perform through the graphical user interface. Unfortunately there are some limitations which seem not quite consistent. I will shortly elaborate on them during the following sections.

### 4.1 The VB Scripts

MindManager not only offers the OpenInterface but also comes with an integrated Visual Basic Scripting engine named Sax. The OpenInterface exposes MindManager objects such as **Application** and **Branch**, methods for object modification and data access as well as object properties.

During data export mainly object properties are interesting since these primarily hold the data a user models and sees. Import works the other round where the map data is read from a file and methods on different objects are called to insert new branches and determine the branches properties (i.e. text, color).

One important point to mention is that XML files are used for persistent data storage. XML combines two big advantages for data exchange:

- It is a cleartext format. Making use of readable markup tags at least provide the possibility for human readers to extract and understand the data.
- The strict syntactic rules combined with a XML schema provide the same readability and understandability to computer applications. Additionally this is a well known and heavily used way for data exchange nowadays and many tools for handling XML data exist.

The above statements lead to an additional necessity: A XML schema. A schema defines the basic and complex datatypes, where complex stands for aggregated basic datatypes, that may occur in a XML file. It also defines the order and relationships of datatypes amongst each other. While XML data without a valid schema (or DTD, Data Type Definition, an alternative to schemas), at its

best may be *wellformed*, that is it is written in valid syntax, a XML document satisfying a schema becomes *valid*. This means that it holds correct data format which allows the computer based evaluation of its content.

What becomes clear, at least at a second glance, is that handling of XML data is not that simple done by hand. Well it may be for XML data creation by using simple recursion tags but mistakes are likely to occur. It definitely is not easy when it comes to the question of validity of a XML document.

Not spending too much thoughts on this point at first hand I than decided to not reinvent the wheel the n-th time but rather rely on already available and tested implementations of XML parsers. It didn't take too long to come across the Microsoft XML parser which perfectly fits into this work because almost any functionality can be used from Visual Basic Scripts too through a COM interface.

The consequence was that little had to be rewritten for the export script, now the code may even look a bit more confusing. The big advantage was the importing part. MS XML is able to validate a XML document against a schema and therefore one can rely on correct data input.

#### 4.1.1 Exporting MindManager data to XML

The implementation can be found in `exportXML.MMScript`.

This file exports a MindManager map to a XML document. Starting from the tree's root node, called the Map Title in MindManager, it recurses down into the maps branches and exports any useful information attached to a node. Since XML data is tree structured by definition no further abstraction or mapping level to retain a maps structure is needed, it all comes for free.

To provide at least a minimum featured user interface I implemented a dialog which allows the user to specify a filename and some options to decide what data should be exported.

- Full Map or branch export  
Allows the user to export the full map or to start export from the currently selected branch item and only export the selected branch and its child branches.
- Export color information  
When selected exports color information such as text, background, font and highlighting colors.
- Export font information  
When selected exports font information for text where available.
- Export text as RTF  
Gives the user a choice about the exported text format for text notes. When selected exports text notes as RTF (Rich Text Format), otherwise uses plain ASCII formatting.



- Encode XML entities

When selected the script does not verbatim export of in-text XML entities but encodes them accordingly. So a < is not mistaken by XML processors for a new tag beginning.

The OpenInterface supports all the necessary methods for querying a branches child branches and selection of one of them. But as life would be too perfect without them there is a little flaw in the design:

Although one assumes there is no distinct difference between the map title and regular branches there sadly exists exactly such a distinction. The map title comes with only few methods and properties. Any method executed or property queried that is not supported is aborted through a COM exception.

Being a novice in the field of Visual Basic I didn't find an easy way of handling these exceptions but luckily there is a property called `Branch.IsTitle` which is a flag indicating whether the queried branch is the map title or not.

#### 4.1.2 Importing XML data to MindManager

The implementation can be found in `importXML.MMScript`.

The import script heavily relies on the XML parser. Since the schema described later on has lots of optional data something like an event based parser would be convenient. I made use of the Visual Basic `select-case` statement. That makes it easy to traverse through a nodes child nodes and emulate a event based parser.

So the script can traverse through the XML node tree, creating the necessary map branches as subnodes occur and set node specific parameter such as their text, color, associated symbols text notes attached to them.

After observing some strange features when importing maps I discovered that MindManager comes with some builtin *artificial intelligence* that automatically adds symbols or modifies other branch properties. At least it does:

- Add an exclamation mark symbol if the branch text ends with one.
- Set the `Branch.PercentageDone` property to 0% when adding the `mmCodeNotDone` symbol (this is shown as an empty square).
- Adding the `mmCodeNotDone` symbol when setting the `Branch.PercentageDone` property to 0%.

And I do believe there are lots more of similar gadgets builtin. This comes in contraproductive when automatically importing Mind Maps because one can assume that the XML file imported either was written by hand or came from another application and in both cases that means that the author specifically did not want to add these markups. This one should remain for the graphical user interface only.

A problem which arose while exporting and importing maps is the placement order of the so called *main branches*. The main branches are the direct child

branches of the map title and are placed and replaced automatically when inserting a main branch. There is a document global property called `Document.MainBranchOrder` but this one only switches between automatic placement or a user defined order accessible only through the graphical user interface.

There is, however, a workaround available which works but it is far from optimal. One can read the x and y coordinates of a branch in the map. When importing one can move a branch to a this specific point while importing to retain the original main branch order. But remember, although it works this is far from optimal.

Another problem arises when using symbols attached to branches. These either use a different coordinate system or some other peculiarities I was not able to find out. But I noticed that some special import sequence can handle this.

1. Import the branch. At first it will be positioned randomly.
2. Import the symbol and attach it to the branch. The alignment beholds a little offset but that's as good as we get.
3. It is now safe to move the whole branch. The symbol will stay attached at the branches side.

### 4.1.3 A Mind Map XML Schema

The definition can be found in `mmp.xsd`.

The definition of a suitable XML schema became necessary because my work should offer a complete guideline for data exchange based on Mind Maps. There already exist some Mind Mapping applications that offer XML export and import functionality, FreeMind even uses a XML file format. The problem that immediately states itself is:

When they all use XML, why did they not agree on a unique schema?

Of course the answer is clear. Even if the developer knew about each others applications there is no documentation of their file formats available. In fact, during my work on this paper I tried to contact the author of FreeMind. Since this application was released under the GPL it was most likely that they revealed their file format.

Then two things happened. First, the original author of FreeMind seems to have retired. The second one was that the FreeMind file format seems to consist of 2 to 3 different elements and about 5 attributes which could easily be identified after taking a look at the source code. God be praised for OpenSource.

So I stood at the beginning again. After having a brief look at the exported files I found that they heavily make use of attributes and almost ignored the existence of elements. As the first point applications should target at internationalization. Storing text in attributes seems somewhat strange since the W3C has already thought about his task and introduced language identifiers to be used as attributes.

However, I have to admit I tried to set up a schema suitable for the MindManager application for that it comes with more different branch properties than

FreeMind. I started after writing the export script and evolved the schema during my further work on the export and import scripts for MindManager. I tried to reuse or inherit from the predefined datatypes whenever this was possible. As an example colors are represented by the hexadecimal representation of their RGB values.

The schema is quite simple and not too comprehensive and for sure gives enough reasons to further elaborate a general schema for Mind Maps. But still it is the only schema I know of at the time of writing.

#### 4.1.4 Data Exchange

What we have got for now are some scripts that allow to export data from MindManager and import the data into MindManager again. I agree that this is not worth to be called data exchange. We also know that FreeMind uses a XML file format and it is easy to guess where this section aims to. A one specific XML format conversion into another XML format almost has to lead to XSL transformations.

XSL transformations (XSLTs) are actually not only handy, they manage these kinds of tasks awfully well. One of XSLTs powerful mechanisms is pattern matching. Knowing about this fact one can imagine the power of these transformations on typically recursively structured data. The implementation breaks down into identifying the source element (or attribut, or ... yeah pattern) and specifying what exactly should happen with the matching nodes.

**The MindManager to FreeMind XSLT** The implementation can be found in `mm2fm.xslt`.

This was my first XSLT and I chose to begin with this one because it meant to filter out data from the richly filled XML file exported from MindManager into the rather spartanic format used by FreeMind. This transformation not only basically converts elements into attributes this is actually all it does. Of course there is one exeception to this rule since FreeMind uses `#RRGGBB` color representation and MindManager represents colors as `BBGRRR`.

**The FreeMind to MindManager XSLT** The implementation can be found in `fm2mm.xslt`.

This transformation is the exact counterpart to the previous transformation. Only it does not match elements but attributes and transformes them into their suitable elements. Again the only exception is the color representation which has to be translated to MindManager format again.

**XSLT prettyprint** The implementation can be found in `pp.xslt`.

Although MS XML does all the data holding and formatting including the actual writing and reading of the files it is somewhat disappointing that it completely failes in the formatting part. XML is designed to be a cleartext, human readable

format but MS XML does neither insert newlines nor does tag intention so that the resulting XML file is in fact unreadable again.

This is not an acceptable situation but luckily other tools like xsltproc and also the XSLT processor coming with XMLSPY provide the functionality. The simplest XSLT transformation is sufficient. There is a XSL function called `<xsl:copy-of/>` which does a full copy of a node including all attributes and subnodes.

Et voila. Applied to the documents root node this function works as a pretty printer XSL transformation which can be applied to any XML file. Using this intermediate step we can produce real human readable XML files while still making use of the MS XML environment.

## 4.2 Passive Visualization

Passive visualization takes Mind Maps away from their standard applications and provides different views of Mind Mapping. Examples include screenshots and similar non-editable Mind Maps.

A completely different aspect is the automatic translation into authoring tools such as Latex or DocBook. I was immediately fascinated from the potential of this aspect. It would mean to combine the power of Mind Mapping in all its aspects with powerful authoring environments.

I decided to target at DocBook as the next step since it supports lots of different output formats including TeX.

### 4.2.1 DocBook

Norm Walsh, author of *DocBook: The Definitive Guide*[DocBook]:

DocBook provides a system for writing structured documents using SGML or XML. It is particularly well-suited to books and papers about computer hardware and software, though it is by no means limited to them.

In short, DocBook is an easy-to-understand and widely used DTD. Dozens of organizations use DocBook for millions of pages of documentation, in various print and online formats, worldwide.

The (traditional) DocBook [DocBook] toolchain starts with a text editor. Tons of different stylesheets exist that can be utilized using an XSLT processor to transform the DocBook source into different output formats. Currently these formats are supported: HTML, HTML Help, Java Help, Man Pages, MIF (Adobe FrameMaker), PDF, PostScript, RTF (Microsoft), TeX, Text, XHTML, XSL Formatting Objects

The DocBook DTD V4.2 comes with 400 elements which is quite a number when first starting off. But there are two main document types: *article* and *book*. The DocBook elements are designed to be human readable and understandable so these document types intend to describe articles and books.

I decided to use the *article* document type because it seems to be the smaller one. A typical *article* document consists of **sections** and **paragraphs**. The principle structure looks like:

```
<article>
  <articleinfo>
    <title>Seminarwork Mind Maps</title>
  </articleinfo>
  <para>.</para>
  <section>
    <title>First Section</title>
    <para>This is a paragraph.</para>
    <section>
      <title>This is a section level 2</title>
      <para>...</para>
    </section>
    <section>...</section>
  </section>
  <section>...</section>
  <section>...</section>
  <section>...</section>
</article>
```

Although many more elements exist the example above describes a complete and valid DocBook *article*.

**The Mind Map XML to DocBook XSLT** The implementation can be found in `mm2docbk.xslt`.

In general the transformation creates one section for each branch and sets the title to the branch text. Any attached text note is inserted as a paragraph into the section's body.

The map title branch is used to build the articles **abstract** section. The `articleinfo` is build from the map title text and the author name. The last step iterates and recurses through the maps nodes and inserts sections and paragraph into the document. No data except the branch and its attached note text is used.

The goal of this transformation is to automatically retrieve a document template from a Mind Map. With an already given structure an author can concentrate on the plain content of the document without the need to think about the overall structure.

**Further Processing** The way to the final PDF files leads over several different paths. I have gone four of them with different results.

First there is a tool called **xmlto** [xmlto] which is a frontend to the XSL toolchain. Depending on the desired output it selects an appropriate stylesheet

and follows the toolchain. For PDF files this means that first a FO file is created which is then rendered using the PassiveTeX environment [passivetex] and **pdfxmltex**, part of the xmltex [xmltex] package.

Working with **db2pdf** [db2pdf] is very similar. The difference is that here the DSSSL toolchain is used. Other possible output formats include RTF, HTML, MIF, JadeTeX, or an XML.

Another package built upon the Xalan [Xalan] and FOP [Fop] projects is the *enovative DocBook Environment* [eDe]. It is designed for the use on MS Windows systems and comes complete with stylesheets and the Java Runtime Environment. The installation procedure is rather easy and batch scripts are provided to keep usage simple.

The third way I have tried was to use LaTeX for further processing. To transform the DocBook XML to LaTeX I have used the *db2latex* [db2latex] stylesheets with very few customizations. There is a custom stylesheet (implemented in `db2latex.xsl`) which allows to set a custom LaTeX preamble. This stylesheet is applied by a XSLT processor (like **xsltproc**) and produces complete LaTeX source that can in turn be compiled with **pdflatex**. Until now it is not possible to automatically add a table of content or similar commands since only the preamble can be modified.

### 4.3 What remains to do

The work that was done during this seminar is by no means complete nor perfect. I will just outline a few ideas that came up during the seminar.

First of all the Mind Map XML Schema may not be suitable for general Mind Maps. It resulted from the attempt to capture all the information the MindManager export script could gather. I admit there may be better representations that abandon certain meta informations and solely concentrate on the content.

The VB Scripts incorporate a flat code structure and heavily rely on the MSXML object. It might be better to encapsulate the MSXML part into custom objects to be more resistant to versioning incompatibilities and the port to other XML parsers and generators would be easier.

A second point is the *feature set* the scripts provide. The entity encoding should include commonly used entities. Also exporting text notes as RTF produces invalid XML files, maybe RTF formatted text should be placed in a CDATA subnode.

The Mind Map XML to DocBook XSLT only supports the style described above. It is desirable to add flexibility regarding the document type and the target elements. At least it would be nice to choose whether a branch title is transformed to a section title or a paragraph. Further I think of some kind of maximum export depth so that the transformation stops inserting sections at a predefined map branch depth. This would also come in very handy for a transformation to a slides document type where only two or three levels of branch titles are used and all the written text is ignored.

Additionally a very nice transformation would lead from DocBook back to Mind

Map XML. Although this seems quite complicated because of the size of the DocBook DTD one could possibly implement a subset such as different section elements, paragraphs and lists. This transformation would enable a rather nice way to automatically get an overview of a document different than the table of contents and also enable the author to restructure a document in a convenient visual way.

At the time of writing (02/2003) a new version 0.6.1 of FreeMind was released. With a new maintainer it now seems that the project is online again. It would be nice to follow new developments and adopt or extend the Mind Map XML Schema and XSLT transformations to incorporate new features introduced to FreeMind.

## References

- [ConceptDraw] <http://www.conceptdraw.com>
- [DocBook] *DocBook: The Definitive Guide* <http://www.docbook.org> , Norm Walsh, 1-56592-580-7.
- [EmMYMap] <http://www.eminec.com>
- [Fop] <http://xml.apache.org/fop/>
- [FreeMind] <http://freemind.sourceforge.net>
- [Inspiration] <http://www.inspiration.com>
- [MSXML] <http://www.microsoft.com/xml>
- [MindManager] <http://www.mindjet.com/>
- [MindMap] *The Mind Map Book* <http://www.mind-map.com> , Barry Buzan and Norman Buzan, 1-56592-580-7.
- [MindMapper] <http://www.mindmapper.de>
- [MyMap] <http://mymap.sourceforge.net>
- [VisMind] <http://www.visual-mind.com>
- [Xalan] <http://xml.apache.org/xalan-j/>
- [db2latex] <http://db2latex.sourceforge.net> , Ramon Casellas.
- [db2pdf] <http://openjade.sourceforge.net>
- [eDe] <http://docbook.e-novative.de/>
- [passivetex] <http://www.tei-c.org.uk/Software/passivetex/>
- [xmlltex] <ftp://ftp.tex.ac.uk/tex-archive/macros/xmlltex/>
- [xmllto] <http://cyberelk.net/tim/xmllto/>