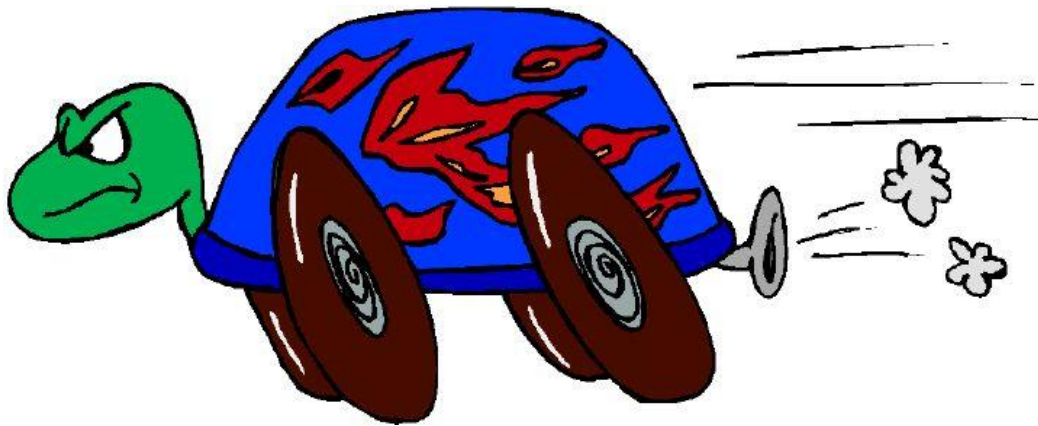# Build eTexts Faster and Better using Rapid Text Construction Techniques

Another applied Electronic Language text

By Dr Elwyn Jenkins Composer of
Electronic Language

To get the most out of this book, you should go to the Text Chunk
Library Information Center where you will receive instructions for
access to the Text Chunk Library.

GOTO: THE TEXT CHUNK LIBRARY INFORMATION CENTER

http://textchunk.info

Verity Intellectual Properties Pty Ltd
Publishers of eBooks and information websites

Is this book for you?

This book is designed to inform a person who writes using a computer at a middle level of competency how to become an expert user of computer programs to compose information heavy eBooks, presentations, websites, and information rich databases.

This book will be particularly useful to people who write user manuals, instructions manuals, information rich texts, 'how to'manuals, and text books.

The 'how to'instructions can empower a person to increase text production when writing from the usual 5,000 to 7,000 words per day to an amazing 25,000 to 50,000 words per day. Increases in productivity will not be experienced until the system described in this book is applied to writing of at least three to four texts. The success of achieving this level of production is in the reuse of previously written text chunks in future writing. This is a powerful concept that will revolutionize your writing.

Through applying the system described in this book, you can now build a range of texts from the one set of data so that you meet the requirements of different niche markets. You can also experience a new level of confidence in writing through extending the knowledge base from which you write each time you set about building new text.

Contents

Foreword

The concepts, insights and techniques of composing eTexts included in this text rise from an understanding of language as it is used in this age of connected computers. This text describes what a writer of texts in this computer age should be thinking about and how she/he can be working to produce texts faster, better and more powerfully. To provide you with some background to my thinking about all of this, please let me walk you through the foundation ideas that have informed this text. All of this thinking is an extension of a study completed over ten years towards the awarding of a PhD. You can read further about these principles by visiting the Technacy Info site. Here are the key underpinnings for a fuller understanding of what "Build eTexts Faster and Better using Rapid Text Construction Techniques is about.

Language

Until a large scale and worldwide network came into place, all language was either one of two forms: spoken or written. Speaking is the oldest form of language used by people around campfires to orally record the history of clans, to pass on the wisdom of the elders and to recite the genealogies so that young people had a sense of where they fitted into the world. However, in Egyptian times, spoken records were not permanent enough to handle commerce that was growing a more international momentum. So, written records became more or less a standard form of recording the commercial activities of the nations. Historians of the time also composed documents, on parchment and clay, to record the momentous events of their time.

Writing as a dominant form of communication did not take a major hold until the time of Plato, however, due to the strength of the oral mode of communication and also due to the fact that writing technologies were not really well developed until just prior to his time. At this time, Plato, being schooled in writing and writing technologies identified that other people were different; he was beginning to lose 'his mind'his way of orally thinking now that he was spending so much time with this literate way of remembering, learning, writing and thinking. The move from oracy to literacy was substantial and has not yet been a universal attainment of humankind. In this century we have perhaps the largest number of people that are literate than have ever been so, however, there are still millions of people who are not literate.

Every human speaks and listens. Not every human has adopted reading and writing, although it is the aim of most every society to produce human beings to read and write. Literacy is seen as a necessity for people to move forward in our world, to develop their fullest potential. Now, however, with the introduction of a worldwide large-scale network, there is a new form of human communication. The dichotomy of speaking and writing has now been added to and transformed so that human language now is not just about oracy and literacy but also now the technologies centering around a computer connected to the Internet. To develop and grow and to have a powerful voice in this age people now need also to be **Technate**. Added to oracy and literacy a person needs to communicate using skills of **technacy**. This means that added to spoken language, and written language there is now electronic language which is composed and decoded on computers for use on computers and is published by computers and transmitted by computers around the world for use in a totally computer environment.

Technacy

In literate activity there are two basic skills: reading and writing. Technacy is far more complex in reach and encompasses four basic skills: operating, composing, simulating and programming. These sets of skills are a highly complex range of skills which it is unlikely that any one person would ever need to command all of the skills. Rather, the range of skills demonstrated by any one person may be highly specialized centering around one or two of the four main skills. Already we see this happening in IT departments where there are maintenance personnel who specialize in operating the machinery, those who simulate and program and those in business who operate and compose.

This book is about building systems in electronic language. A system is a meaning generating environment constructed from two or more interactive programs, that share at least files, and which can be used to build structures for information sharing. This involves the skills of:

- operating programs,
- composing objects and files,
- simulating large information products such as websites, ebooks and other large texts.

Orate and literate activity shares a common study that identifies the structure of language in use. The study of grammar is a study that analyses clauses, identifying the range of potential meaning structures that are available to the speaker or writer in constructing language. Grammar identifies the ways people can get things done through exploiting the meaning options that are available in the language. Electronic Language requires a different study than Grammar to understand how this language is structured. The study of **Synthesis** provides us with a grammar-like view as to the options of meaning making that are available when a human is composing electronic language texts.
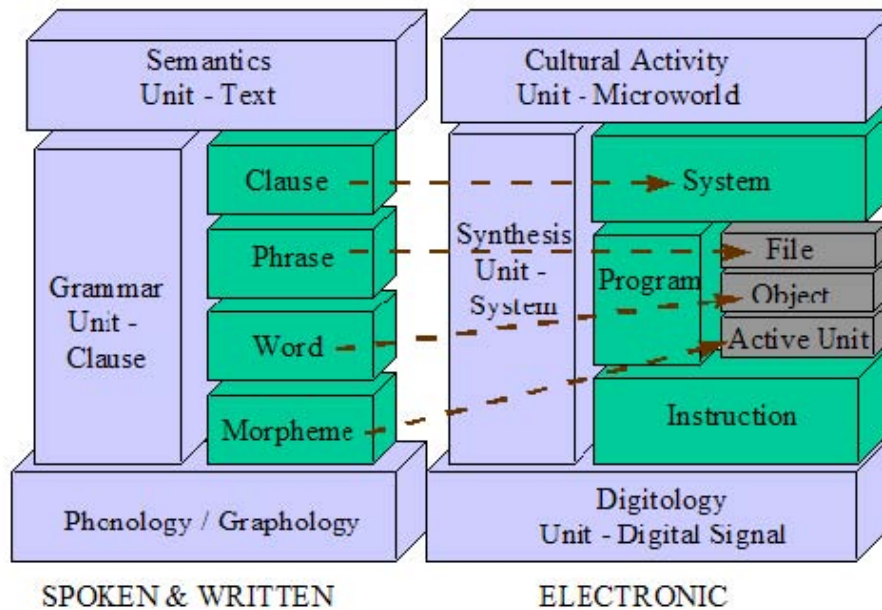
Electronic language is a meta-language that is primarily about embedding all other forms of language in meaning systems for use in the totally electronic environment. What can we say in electronic language? We can say most anything that can be said in spoken language, written language combined with graphic communication and packaged into bundles that can be apprehended on the internet. At the level of system we can make meanings, using the meta-language of electronic textualisation, that center on what we can do, shall do and intend to do with text, graphics, and recordings. This is a language of simulation that also involves composing and programming.

Synthesis

Synthesis is the resource, similar to grammar in written and spoken language, for *computing* meanings; it centrally identifies the way to get things done through exploiting the meaning options that are available in the language. Just as grammar primarily considers the structure of a clause and its various components, synthesis considers a *system* and its components, programs (files objects and active units) and instructions. Each of these components are defined as:

1   Instruction An instruction is a command or an action that is pre-written to form a program, and also includes the command or action that a computer user must issue in order to direct a program to perform a job. There are therefore **instructions that are programmed** and there are **instructions that are added by the user** at the time of operation. Altogether, a series of instructions combined together form a program.

2   Program A program is composed of instructions that are pre-programmed, as well as added by a user at the time of operation. Several programs that are combined together to get a job done form a system. Programs can be subdivided into files, objects and active units. Files are the devices that are stored on a disk and fed into active memory that are a record of the pre-programmed instructions, and also may involve instructions issued by the operator in this session or in previous sessions. Files are composed of objects and objects are composed of active units.

3   System A system is composed of **a series of programs that are combined in a particular way to share files between those programs to get a specific job done** as specified by the composer. For example, the three programs PersonalBrain 2.01, MS Word and Adobe Acrobat can be combined into a system. Through a composer adding instructions to these programs, files can be created, defined in .RTF (Rich Text Format), and stored in PersonalBrain, combined in MS Word in specific ways, and a file issued for conversion into .PDF (Personal Document Format) for display as a eBook in Adobe Acrobat Reader. A specific system would include the actual instructions to be added to a program suite, a definition of how the programs are to be used to compose files, and how those composed files are used to build a particular information product.

In spoken and written language, we test any language we see or hear by comparing this language with other models we see and hear. We often say, "Oh that is not grammatical" meaning that it does not follow the options or resources available in grammar. When we work with Synthesis, the driving need is not to be grammatical but to be **efficient**. The overriding social requirement of computing devices is that we get jobs done efficiently, meaning faster, better and cheaper. That is, faster for the human who has a cost of operation and not necessarily faster for the computer; the computer may take processing time far beyond the processing time of a human, but since the human can do other things, the overall effect is for greater human efficiency. However, the ultimate in synthesis is where both human and computer are empowered to get a task done that is hugely efficient meaning that both human and computer can get a job done faster, better and cheaper.

An Example of a System

The following diagram outlines a particular system designed to create re-useable text elements and provide an automated updating system for production of ebooks. The objective of this system is to build ebook-text efficiently, meaning that an entire book can be produced with as little time as possible of human time and computer time as possible and obtain an extraordinary result.

With specific instructions an operator can form a range of files in PersonalBrain that can be combined to form a document in MS Word and then converted to form a .PDF document for use in Adobe Acrobat. Since the instructions used by the operator compose the text in a particular way, the following is possible:
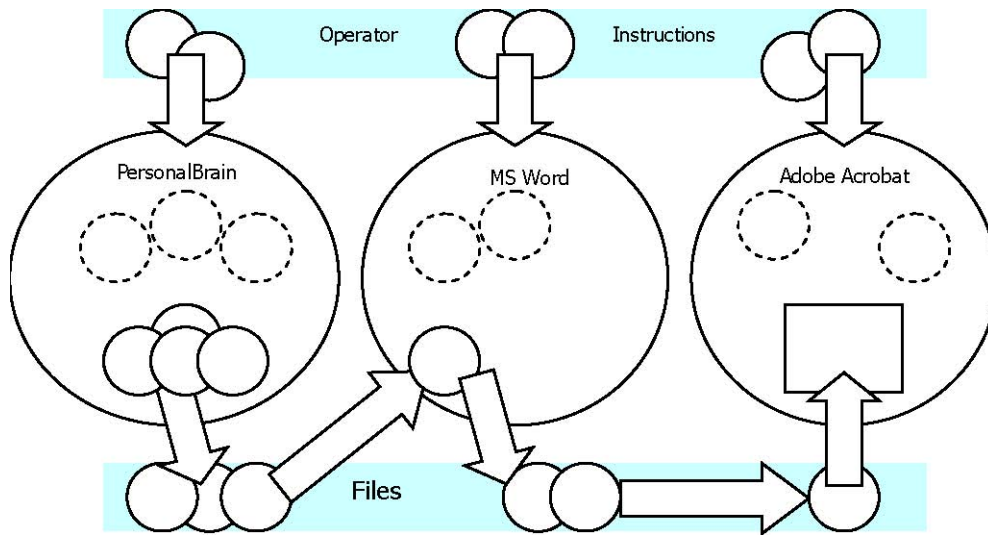
1   Each of the files in PersonalBrain can be updated and automatically the subsequent files in the system can be updated;

2   The files in PersonalBrain can be re-combined into different order to form a different document in MS Word and Adobe Acrobat;

3   A subset of all available files can be combined to form a new MS Word Document and subsequently form a different Adobe Acrobat Document;

4   Files in PersonalBrain can be combined with files in other Brains to form a new document in MS Word and then compose a new Adobe Acrobat document;

5   The Brain composed with these files can be published for other people to use those files in a similar manner to compose new texts.

This book

In this book, the instructions, program actions, resulting files and overall system operation that is described is a system designed by its composer using the principles of synthesis to achieve an efficient result in generating ebook text. This is an analysis of a particular system operating on the author's computer for production of this and other eTexts; additional examples are also taken from other synthetic projects devised by the author to illustrate additional potentialities in the range of technate activity.

1. The Composing Suite

As a composer of electronic language, your task is to use the resources available to you, the resources of synthesis, to build a system that efficiently gets a job done. To set out in building a system, you need to have the following organized:

1   Determine what the **scope of the job** you are about to do entails;
2   **Identify the programs that are available** on your computer, across the Internet, or on sale elsewhere, that can be used in combination to get your job done;
3   Work out the **method of operating each program**, the necessary instructions and parameters you are going to give each program, and the method of **interchange between each program** to gain the greatest efficiency in getting the job done.

This book is both an example of how one composer went about creating an efficient system for generating text for ebooks, and it is also instructive of how synthesis can be used to get other jobs done efficiently that you may encounter as a composer of electronic texts.

1.1 What this book is about

The author wanted to make his work of generating ebook texts more efficient. This book is about the system the author composed in order to make ebook text generation more efficient:

1   When generating a single text;
2   When generating subsequent texts;
3   To generate new texts that previously would have been not considered.

When generating a single text the author's method of writing involved:
1   Collecting ideas for the text through online research, direct book research, original thought, and drawing together a series of arguments and a conclusion;
2   Forming these ideas or thoughts into a logical progression;
3   Developing an overall structure for the text –chapter headings, subheadings, and contents of texts under each, usually in an arrangement of no more than three to four levels of headings (1., 1.,1 1.1.1, being three levels of headings);
4   Filling in each section under each heading with the ideas for that section;
5   Writing out in full the ideas under each heading.

Inefficient activity was identified in the following areas:
•   Brainstorming, mind mapping and/or collection of references for locations across the Internet was done in one or two programs, such as Ygnius which involved typing ideas into a map. This text was not available for electronic copying or use in the word processor. Thus, even though ideas were recorded, those ideas of thoughts were all re-typed at a later date.
•   Often after all material was written, the progression of ideas did not seem to be effective. Thus, the material was re-organized often with a lot of reformatting needed to cater for the changes of emphasis of the ideas or thoughts.
•   Through the writing process, with changes in format, alterations in the styles used, and in chopping and changing text, a large number of styles were created in word, most of which could be eliminated. Through eliminating these, re-formatting would have to be re-done, or through not eliminating these, the conversion to a .PDF format would stall and cause difficulties.

When generating subsequent texts that were either updates of the first ebook, or that were another ebook based on the first, there was always text that could be re-used. In efficient activities involved in building these subsequent books included:
□   Editing the original book to delete the text not now required;
□   Re-formatting the new book to reflect the new logical progression;

☐ Eliminating the styles not now required and obtaining a good clean document for production of the .PDF document.

Generating new texts out of the current texts available often did not occur to the author simply because there was not way to make connections between the various discrete books that were produced.

This book describes a system that was built to erase these inefficient activities, and how that now forms a system through which the author generates multiple texts using this system, effectively databasing the entire production of text he produces to form reusable text objects that can be effectively and efficiently inserted into any text in production. This forms a rapid text development system that can be used to generate new texts of more than 50 to 100 pages in length in less than three days work.

1.1.1 Why not one program?

The whole premise of the concept of 'System' assumes that there will never be a single program that can be adopted by the composer to achieve the composer's end goals. Why is it not possible to find a single good program that can get the job done as required by the composer?

There are several good reasons why it will never be possible to build a single program that does everything. First we must consider the nature of the problem we are solving; the way I think about the activity I am now embarking on may be quite different to the way you may think about the problem. Different thinking styles require different program activities. For example, you may prefer a Mind Mapping program that holds all thoughts in the on-screen picture. Personally, I do not really care about what other thoughts there may be, as long as I can see the relevant and immediate thoughts. The programmers of MS Word never envisioned anything remotely like PersonalBrain being a part of 'the writing process'; their style of thinking about how to write is a different concept to the one I am working with and elaborating. The concept of re-usable text chunks never entered their thinking as their concept has been based on 'original' thought as a writing method; so to them an outliner is an appropriate method of textual development. So our different styles of thinking about writing cause there to be different pre-set elements within a program.

A program is a program development production team's best guess at what people will need when they embark on a writing project. They have studied a wide range of writing contexts and have made the program as general as possible with as wider range of tools as possible for those people who will embark upon writing. Generally, MS Word is built for an office where business texts are generated. Simply, their concerns are not those of mine in generating text, and I do not expect them to have the exact concerns. There is a complete meta-language of electronic meaning making here, and I do not expect one team to embed the whole language in one package.

Second, due to the nature of programming, it is better that the programming team focuses on one major activity of a program and gets that right, rather than trying to solve all problems in the one program. There is a complex enough job for a programming team to get one major thing right let alone trying to solve all electronic language problems. For example, databasing text chunks has been a quest of this author for ten years. Personally, I have devised fifteen different ways of doing this, even to the point of having my own programming team develop a concept for me. It took another person somewhere else who has invested millions of dollars in a paradigm or way of thinking to actually come up with the solution. Solving problems in electronic language is a huge investment. So with TheBrain solving this problem, I do not want them to get distracted with say, trying to develop a book formatting system. It is better that they focus their investment on improving the PersonalBrain program.

Third, for people to solve problems in electronic language, it is better that there are thousands of programs that have interchangeable data that can empower people like me to

Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- ➢ HTML (Free /Available to everyone)

- ➢ PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)

- ➢ Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below