HORUS and the SYNTOPICON

System Architecture and Algorithms : An Introduction

(or, How to build a Brain and Live with it – a Brief Guide)

Martin Dudziak
Chief Strategist, New Vision, Inc.

Version 1.0
19.August.2003

TABLE OF CONTENTS

Introduction

This document is an introduction that is both high-level and semi-detailed, spanning two different areas of interest. First, it addresses the general structure and design of the HORUS Learning System and the Syntopicon's place within it and as a component within another system that is intended to be both software-based and a physical in-print series of books. Second, it speaks to the more detailed definitions of key components including algorithms within the HORUS system.

The Syntopicon is one of the principle components of the IDEAS Encyclopedia. As such it exists in both in-print and internet versions and these are tightly coupled. Together they are resources for readers and subscribers to use by itself or in conjunction with other components of the IDEAS Encyclopedia and they are maintained by an editorial team responsible for any modifications and new editions. [1]

The IDEAS Syntopicon is described in brief as follows (8/2003):

> A three-volume collection of fundamental themes and topics common to all peoples and cultures. Approximately 160 different topics are presented in the form of essays, outlines, and references.
>
> The internet version is envisioned as an expanded version of the printed version, in terms of additional media and also many links and annotations.
>
> There are @ 160 themas. See http://futures.zonus.ru/list.ghtml?ID=157&STAZ=fffff39174e030faf4f2965b3471d8c4
> for a preliminary list compiled in 2002 – but this is not a final set and the software in this prototype will not be used in the final version.
>
> Each thema has an essay that has several standard components:
> a) intro – more than an abstract, generally one page or at most two.
> b) main body – topical essays - comparable to what is found in the Britannica Syntopicon ("Great Books" series)
> c) outline of topics – comparable to what is found in the Britannica Syntopicon but expanded and including more variety, more non-Western topics, and (technically speaking) operating more from the standpoint of Topic maps and Semantic Nets and the Syntopicon concepts of the HORUS system (http://fortehorzions.com/horus/v1)
> c) references to articles in the Academopedia and in the Great Works
> d) examples (primarily people, historical situations) with the references to articles on these people or situations
>
> Each volume is approx. 400 - 500 pages

The Syntopicon is also the name assigned to the knowledge base and automated learning repository of information that comprises the core of the HORUS Learning System. For distinction, the term Syntopicon-H is used to denote this body of software and data which is a component of HORUS.

HORUS (Hierarchically Organized Reasoning and Understanding System) is a computational system for maintaining a library of knowledge about objects and their relations that can be used by a wide variety of humans and other computational systems for understanding the meanings, uses, and relationships between different concepts, terms, physical objects, and media including objects that are text, audio, image, video, and other hypermedia (web-based) entities. It is a system for knowledge acquisition, hypothesis generation, inference and learning and is designed to operate with a highly interactive, multi-user, internet-based environment that will be accessible to a diverse community of users (public-access or membership basis) for search, comparison, and evaluation of many media types.

---

[1] The term "syntopicon" appears to have originated with Mortimer Adler and the development of encyclopedia Britannica's "Great Books of the Western World" series during the late 1940's and is now a fairly widely used term referring to similar collections of themes, ideas, and concepts.

Users collaboratively build and develop the content of the knowledge bases therein, creating resources that can be employed by both human users and "artificially intelligent" agents for conducting expert-level searches, evaluations, comparisons, data mining, content extraction and summarization on a variety of media. Such media will include documents, articles and other text-based objects but also non-text media as well including images, audio and video. An important and in fact the central component of HORUS is a complex structure of functions and databases known as the Syntopicon which has aspects of its data structure and functionality in common with semantic networks, topic maps, dictionaries, thesauruses, and encyclopedias.

HORUS is designed to be a stand-alone application that runs on a server and is accessible to its users (human and automaton) via the internet. It is also intended to be used in a modular fashion as a component of other systems that may be specific to particular application areas and topics. In this sense, an e-learning collaboratory system can have a HORUS module built into it for its interactive users, as can a healthcare disease management system, or a counter-terrorism security system, or a publishing network such as TransPrint.

Insofar as HORUS is used as a stand-alone system, it can be in a closed environment (intranet or user community restricted to a certain membership) or it can be open and public. There are important reasons for both methods of operation. As a completely open system, HORUS can serve to develop a very culturally diverse knowledge base that as it grows over time, with its own autonomous learning capabilities also growing over time, will have the opportunity to become a truly power knowledge resource. The value of the closed or limited user-audience implementations are that the development of the knowledge base and the methods of relationship-building among the Syntopicon-H elements can be much more controlled and focused to the needs of a particular user community.

Within the development of the IDEAS Syntopicon, plans are currently to make use of the early versions of HORUS for building the topical outlines and cross-referencing between the @ 160 themas that will constitute the IDEAS Syntopicon.

Syntopicon-H and Its Structure and Use within HORUS

HORUS collects and relates knowledge about objects using information given by users (humans and automatons[2]) about those objects and then processing this information, in the form of relations that are definitions, associations, and inferences, to produce modifications in the knowledge base, such modifications being in terms of the same types of relations.

TYPES

The objects can be of many different fundamental types. Currently there are twenty. These are described in the following table which describes what is the object and how it will be employed, apart from the standard format of creating definitions, associations and inferences (described below), within HORUS: [3]

Object Type	Description	Computational Usage
concept	a word or phrase; an idea	text entries
symbol	something that the user/definer intends to be understood and used as a symbol for something else	text entries plus optional graphic example
sign	a sign, similar to a symbol	text entries plus optional graphic example
reference	a reference to some other element in the Syntopicon	text entries
document	a conventional document (HTML, PDF, DOC, PS, etc.)	the document is submitted either directly or by URL
memo	a text document that is understood and intended to be used as a memo, generally less than two pages and referring to some other object	the document is submitted either directly or by URL, but usually in direct form
quote	a text quotation that is understood and intended to be used as a quotation	text entries
image	a conventional image (JPG, GIF, PNG, etc.)	the document is submitted either directly or by URL but usually in direct form
video	a conventional video file	the document is submitted either directly or by URL
audio	a conventional audio file	the document is submitted either directly or by URL
equation	a mathematical equation, regardless of its representational format	the document is submitted either directly or by URL but usually in direct form
algorithm	an algorithm, regardless of its representational format	the document is submitted either directly or by URL but usually in direct form
code	a body of software code	the document is submitted either directly or by URL but usually in direct form

[2] An automaton within HORUS is a software agent, designed according to certain protocols and known also as a BINAR (Biological/binary Intelligent Neural Adaptive Recognizer), which acts using "AI" (artificial-intelligence) algorithms to perform tasks that are like what a human user might perform

[3] More details will be provided in a separate paper on the nature of HORUS Objects

URL (docmt) (“URLD”)	a standard URL where a typical page is expected	text entries
URL (stream/cam) “”URLC”)	a URL to a webcam or stream	text entries
URL (store/market) (“URLM”)	a URL to an e-store site	text entries
URL (chat/forum) (“URLF”)	a URL to a chat or forum site	text entries
URL (search engine) (“URLE”)	a URL to a search engine	text entries
physical object (“phig”)	a reference to a distinct physical object	text entries plus optional other media (e.g., images)
virtual object (“virt”)	a reference to a distinct virtual object	text entries plus optional other media (e.g., images)

STARTING POINT

There is no formal system that can be complete in the sense of having no undecidable propositions. This is the proof as the heart of Goedel’s Theorem and we work from the basis of this and other demonstrations along the same lines.

From another perspective, there is no dictionary without some definitions that will be problematic and disturbing perhaps to some readers, but ultimately a good dictionary works hand in hand with a good primary learning of very basic words and concepts that must be in some sense “fundamentals” – for human individuals, these are usually things that are learned at an early age and about which there is not a great deal of variance or dispute. Note, however, that while we cannot yet establish a formal proof, it certainly appears to be the case that there is not a single “basic concept” learned in early childhood that cannot acquire a variety of different and even contradictory meanings later in life, some of which are problematic, others ironic or humorous, and in any case the clear indicator of the imaginary and creative faculties of the human mind.

HORUS begins with a set of basics, really fundamental concepts that enable the system to have a “head start” compared to the learning level of infants with respect to formal language, although this proposition in itself can certainly be the trigger for a number of interesting debates among both psychologists, linguistics, and cognitive scientists! Our aim in the first version of HORUS is, however, not to become bogged down in a tedious and difficult academic study but rather to defer such interesting examinations until later when we have a robust and working system that is growing daily with new information and knowledge. Furthermore, one of our primary objectives is to have a system that can be put to work in the “real world” of commerce and education, in several forms, in order to not only sustain the operation of designing, building and operating such learning systems but to put the engines and databases to the most challenging set of tests possible for any information system – can it be used practically and gainfully by ordinary people and in ordinary everyday applications?

THEMATONS

The Starting Point for HORUS is found in the thematons,[4] a set of potentially several thousand base concepts forming the foundation of the Syntopicon-H knowledge base within the system. This set (also called the T-Set) is expected to start with at least 800 entries and cap off at approximately 6,000 and it is international in nature – not only English.

Thematons are entered into HORUS like any other object but they are handled somewhat uniquely within the system because they form the basic “semantic skeleton” by which all the other elements and their relations with one another are integrated and assembled into what we view as an “organic whole.” There is no escaping the fact that by so doing, creating a particular skeleton as it were, we are creating a very particular

[4] from “thema” or “theme”

"knowledge geometry" that is going to influence the way the whole system behaves henceforth. This is unavoidable.

"We have to start somewhere with some structure"

Another way of putting the matter is like this: "every organism and building (within certain broad classes, like humans, other mammals, even insects, leaves for that matter, and most buildings constructed by humans) has some skeletal structure that holds up everything else and to some extent 'organizes' the whole structure."

Thematons are not limited to only abstract concepts, even though it may appear to be so from looking at the list composed for the Syntopicon within IDEAS (cf., http://fortehorizons.com/ideas). Thematons can include any soundly "basic" terms and it really depends upon the concensus of the editorial team that is managing this aspect of HORUS development.

What sets apart thematons in a functional and information-processing manner is that within HORUS there are numerical assignments made to each element contained in Syntopicon-H. The thematons are manually assigned numeric tags that are of a certain range of values (described further below), and then these values are employed in the assignment of numeric tags for all other elements that are introduced to the system, either interactively by humans or non-interactively by automatons.

ELEMENTS

Everything that goes into HORUS is an element. An object of any of the twenty base types can become an element in the Syntopicon-H knowledge base, and all elements are represented in two formats within Syntopicon-H – in their textual, strong-based form within a relational database (the RDB, where all the editing and updating occurs) and in a numeric matrix database (NDB) that is how everything gets used for the construction of new relations, the execution of inferences and the answering of questions.

All elements are either single words, phrases or other types of strings (e.g., strings based upon URLs and filenames). They are entered into the "name" field of the ***element*** table. The system does not care if something is a single word or a phrase. Phrases are also treated as "words" in the system; i.e., a particularly common phrase may be entered and it will be treated as a block (e.g., "rat's nest" or "young buck" or "needle in a haystack") separately and distinctly – another entry in the Syntopicon-H, handled no differently than a single word (e.g., rat, nest, needle).

RELATIONS - "The Meaning is in the Use"

HORUS is based upon this fundamental concept which derives from a long tradition in philosophy and science and the actual quotation is attributed mainly to Ludwig Wittgenstein who employed it in his Philosophical Investigations.[5] Following this logic, HORUS is designed with RELATIONS and they are of three types:

Definitions
Associations
Inferences

Relations themselves can be considered to be functions that together (all relations for a given element) create the complete function of the element, and in the numerical processing within the knowledge base, this is reflected in a set of values being generated for the element, all of which play a role in influencing the values of other elements and the interactions among elements. This gives to HORUS and its knowledge engineering process the character of not only a very large and complex Network but also a Cellular Automata System.[6]

[5] Wittgenstein, Ludwig, Philosophical Investigations, XXXXXX [give publ. info and URL to full version – see EB]

Each category of relation has three (3) varieties.

Relation	Subtype	Description
Definition	S (subject)	a word or phrase denoting what-this-is; the most basic type of definition and relation *example: (inhabitant of Amazon rainforest)[s]*
	S-A (subject-attribute)	the subject is like the S-type and the attribute is a modifier, typically an adjective or adjectival phrase *example: (1930's era passenger)[a] (airship)[s]*
	S-R-O (subject-relator-object)	the subject is like the S-type, the relator is something that expresses a connection, action or identity, with the object, in a classical subject-verb-object manner *example: (beaver)[s] (cuts)[r] (trees)[o]*
Association	S (subject)	the subject indicates the entity that is being associated with this element, not the element itself; this is the most basic type of association *example: {assuming element is "Iraq"} (oil)[s]*
	S-A (subject-attribute)	the subject is the entity being associated with this element (again, not the element itself) and the attribute is a modifier, typically an adjective or adjectival phrase *example: {assuming element is "Kuwait"} (Arabic civilization)[s] (modernizing cosmopolitan) [a]*
	S-R-O (subject-relator-object)	the subject is like the S-type association, the relator is an operator, verb-like in function, describing how this associated entity relates to the third part, the object *example: {assuming element is "Zeppelin"} (dirigible airship)[s] (powered-by)[r] (diesel propeller engines)[o]*
Inference	D (deduction)	two parts – assertion clause (IF …) consequence clause (THEN …) *example:*
	I (induction)	three parts – basic condition secondary condition consequence *example:*
	A (abduction)	three parts – primary assertion secondary clause hypothesis (implication) *example:*

All elements receive such relations and they are input either interactively by human users or autonomously by BINAR agents. While in principle a BINAR automaton can create a new element "from scratch," this is futuristic at the moment. BINARs will edit and modify the knowledge bases, but generally new element

[6] As a CA, HORUS is in particular a CLAN or Cellular Local Area Net system (no relation to LANs in the IT and computer networking world) where there are many subnets or clusters that operate internally and then interact at a higher level with other encompassing subnets, and so on up an indefinitely divisible and fractal-like scale. Thus, the "hierarchical" aspect in the name, HORUS.

creation is done only by humans using an interface that is similar to what is accessible in the HORUS prototype (cf. http://fortehorizons.com/horus/v1 or the in-progress work on the SourceForge project site).

At the time of element creation and at any time thereafter, there is no requirement for all of the relations for a given element to be completed or to have any values, at all. A user can simply introduce a new element and its type (these two fields are mandatory) and go on. This creates an entry in the system, starting with the relational database (RDB), and there will even be an entry in the NDB when that is updated, but this element will not have any relations and thus no connections with the rest of the HORUS world for as long as it is left alone. However (!) users can make relationships from other elements to this element.

These points may seem difficult to fathom at first but it is all part of the overall philosophy of HORUS that knowledge and understanding are built gradually, piecemeal, ad hoc, and with some degree of self-organization. It is OK to have incomplete ideas in HORUS, just like in "real life" – even very incomplete and fuzzy ideas. Contradictions and non-sequiturs are allowed, too! We build up reasoning by association, implication, touch-and-go, feely-touchy thinking. Gradually the community of the minds involved with HORUS, just like the community of mindsets, ideas, dispositions, emotions, and ways of using thoughts and concepts inside our individual minds, evolves and self-organizes a more complete and integrated whole, and this becomes stronger and stronger for influencing whatever comes next in the way of new thoughts and relation-building among elements that have already been defined, associated, or inferred.

Another significant point is that as elements are entered into the RDB, no error checking is performed except for elementary syntax and spelling at the GUI level. All such processing about accuracy in definitions, associations, and inferences belongs in a deeper and later level of the knowledge building process,[2] both interactive and automated. We correct things and reshape our minds when we are "told" by different parts of our conscious reasoning process (including the influence of moods, feelings, and intuitions, a different matter in a way but still related to the process of learning) or through teachers, mentors, and those other entities that in some fashion or another force us to change.

RDB – the Relational Database

The RDB is maintained within a MySQL database but designed to accommodate changes to other database engines. ZOPE is being considered as an alternative. It is not envisioned that the RDB will need the overhead and transaction-intensive features of an Oracle product, for instance.

The RDB data structure and entity relationship diagrams are provided in separate design documents. Presently, one should consult the file install_syntopicon_tables.php.

Within the RDB there are string fields for the most part. The data that goes into the RDB comes from the form entries completed by the user doing adds, edits and deletes to elements and their relations. There is also computed data that is mainly for managing the database and for faster query and retrieval operations.

The only special value that pertains directly to the NDB and the core of HORUS operations is the ***mnum*** (matrix_number) field within the ***element*** table. This number is the number assigned to that element, not as a standard database key ID but as what will be used for the history of this element's life in HORUS and for all of the relation-centric operations.

Differences between Thematons and Non-Thematons re: mnum

Here is where the differences really show up – in the mnum.

Thematons are always given a matrix identifier (mnum) that is a real number and it is separated by all other mnums by a factor of 100.0. The initial mnum used = 100.0, 0.0 being reserved [7] (thus, 100.0, 200.0, …).

[7] Not so much at present for anything special but just to avoid confusion and error in some future time.

Given our initial projections on the size of the thematon component of the Syntopicon-H, we can expect the highest value for a thematon (and thus, we shall see, for other elements) to be in the vicinity of 600,000, certainly under 10^6.

Originally it was thought to group different thematons together and thus have some type of clustering within the T-Set. However, this was found to introduce too many arbitrary clusterings and also separations between thematon mnums, because of the fact that there are simply many <u>uses</u> to these "core ideas" and it would disaffect and disturb the whole equilibrium that is a prime directive for the Syntopicon (in both its forms), to be as open and balanced and useful to as many people, applications and interpretations as possible.

Thus, the assignment of mnums to thematons is arbitrary and simply in the sequence in which they occur, with plans to have some type of reorganization and sorting so that they are all in some semblance of alphabetical order (the "some sort of" and that vagueness is because of the intention to include not only English (Latin) words, so there will be some arbitrariness to how several letters from other alphabets are "fit into" the scheme from "A to Z."

Each non-thematon element among all the other entries (i.e., the vast majority of what can be expected to be in any HORUS system's Syntopicon, upwards of 90 – 99.9%) is associated with a numeric matrix value mnum that places it in some default "starting" relationship with a thematon. This mnum has an initial calculated value and this can be expected to change even radically over the history of the HORUS system. The mnum is calculated (initially) by the following:

$$m_n = m_T + v$$

where

m_n = mnum[some new non-thematon element]
m_T = mnum[some relatively close associable thematon]
v = initial_variance

The initial variance v is calculated on the basis of how "distant" the new element is from the base thematon element that has been roughly associated to be related to it and vice versa. None of this is intended to be rigid or exact, since everything will change during the regular updating of the NDB from the RDB, and this will affect the mnums of non-thematon elements in the RDB accordingly.

This value v is a decimal number (1.1, 5.43, 26.0, 87.65437…). How v is calculated automatically is still under development. Presently it is left as a semi-manual process – the user indicates a qualitative relationship with the initial associated thematon, and that is translated into a number. However, the deeper algorithm will take into account automatically all of the relations that are established with the element and use those to compute v, taking into account any definer-assigned weights that are in the RDB (provided at create/edit time), plus relations among those related elements, to a second degree.

It is very important to remember that the RDB records for each element are built, manually or automatically, from the outside looking in, so to speak – a user has access to the whole RDB potentially (generally only a subset at a time, for practicality, at least for the human users) and builds a new element and its relations. But this process of creation (or editing or deleting) does <u>not</u> affect <u>at that time</u> any other single element in the entire HORUS system, nor does it immediately affect anything at all in the NDB which is the realtime "workhorse" for all inquiries, lookups, searches, and exploratory reasoning tasks that HORUS may be doing.

<u>**NDB – the Numeric Matrix Database**</u>

The NDB values that are assigned to the elements and their relations, and the fairly simple (computationally speaking) combinations and relationships and operations among these numbers is what yields the various results used within HORUS for assisting search, exploration, and inference tasks.

Each element has an mnum. Each relation is in terms of other elements but many of these will be completely undefined. Many of the relations are not explicitly in terms of elements in the Syntopicon, but in the processing of the RDB, each relation of a new element is examined for what elements can be identified and associated. This involves a very simple natural language processing function that employs dictionary look-up functions to build from each relation a ***serl*** or simplified-element relation. It is estimated that this serl-creation process is likely to be the most computationally intensive aspect of processing a given new element from the RDB into the NDB or for processing any future reference or search for a given element, because of the potentially large number of natural-language parsing operations to be performed (all depending of course on how the new element in the RDB has been described by its creator/editor; it could also be a fairly simple task)

The ***serl*** is of the same structure as the basic relation as given in the RDB by the creator/editor (human or automaton).[8] However it is simpler and only in terms of elements, and with calculated measures of closeness for those elements chosen.

Based upon the elements used, and in particular for S-R-O type definitions, S-R-O associations, and for inferences, there is a mapping of semantic weight values ω_i to the serls; the weights ω_i in turn become attributes of the serls for their use in the computation of a set of lists known as the R-Set. This is a very central facet for the operation of the HORUS system and not only pertaining to the internals of the NDB.

Thus a ***serl*** has the general forms:

(ωe) for an S-type definition or association
(ωe) (ωe) for an S-A type definition or association
(ωe) (ωe) (ωe) for an S-R-O type definition or association
and more complex forms [being developed] for inferences [be patient!]

The R-Set is a list of lists for each element. Each list is connected with a particular relation (and serl) and has a data structure that is particular to the type of relation with which it corresponds. Each list contains a list of <u>tuples</u> that each have two or more numbers, each of which include one mnum of some element that is in some probabilistic relationship with the "parent" or "subject" element to whom the list and the whole R-Set belong and minimally one probabilistic weighting value.

The R-Set may be understand as follows:

R-Set = {a, b, c, …} where each element in this set = {p1, p2, p3, p4, …}

and where p_i = t, a tuple of variant membership depending on the nature of the sets a, b, c.

A table can be created to understand this mapping. Assuming there are in this universe only six types of lists, we may have:

ListType	TupleSize	Description
a	2	weight (wt), element (e)
b	4	wt1, e1, wt2, e2

[8] Most of the relations for elements that will be introduced (created) or modified (edited) through BINARs can be expected to be pretty much straight se_relations because BINARs, though we plan for them to be as smart-looking and smart-acting as possible, are not likely, or required, to generate the variety of language used by humans. BINARs will be smart in an efficient sense, but one should not expect them to be poets.

c	3	wt1, e1, some-other-modulator
d	3	wt1, wt2, e
e	4	wt1, e1, e2, e3
f	2	wt1, e

Now when the R-Set for a given element e has been assembled, this R-Set plus the mnum and the serl form the suite of data for that element e within the NDB. All of this is in numeric form. This means two things to the saavy programmer, server admin and end user – first, speed in performance, and second, compactness in size. Although the NDB can be expected to be very big as it grows, especially in a non-specialized mode, it can be partititioned logically and computationally.

The NDB can be archived on disk and run operationally in diskless mode using only RAM on a server.

This means even greater speed. This means basically having your brains where you want them – in the fastest part of the computer (excepting on-CPU cache).

The value of the NDB, however, is not only in speed. By transforming the relations of the RDB into the numeric sets of the NDB, HORUS brings into one place for each element e a potentially vast array of relations that have not been considered by the creator/editor, and even beyond those that may be explicitly implied by the inferences that may be in the relations for that given element.

At this point we must consider multiple and conflict definitions and also the mapping of relations from elements to each other as a result of the introduction of new elements into the system. Consider a new element e101 that has certain relations with elements e1, e5, e14, e22, e37, and e86. Let it be that those in turn have relations with a variety of elements. Some of these may, as a result of the relations and weights with e101, have relations with things that are very strongly mapping with "identity" type relations (e.g., "is a something_like_e") that involve the element e. This brings about the generation of new serls within the affected elements (e.g., e1, e14, e86) and those serls now become part of the NDB for those elements. All of this is occurring within the structure of the NDB. (Note that those new serls are <u>not</u> back-entered into the RDB, at least not within the initial version 1.0 of HORUS.)

Multiple, Conflicting and Mapped Relations

All elements except the thematons can have multiple definition-use values assigned to them. So the word and concept "bear" might have associated with it, as an example, things to do with animals in the forest or carrying things. The clarifications will work themselves out over time and usage.

MORE
- - - - - - -
The thematons do have multiple definitions and uses, of course, so how is this handled? This is done through the relational operators, and the definition-use for a thematon is singular; i.e., only one, taken as the "primary" or most common usage. While this may seem "dangerous", the relational operators and the general statistical behavior of the system help balance everything out in the end.

This section is not finished!!!!

NEXT (new section, untitled)

Once the NDB has been updated with new edits to the RDB, an event that will occur on a scheduled basis, driven by system demands and the number of changes into the system (meaning through the RDB, mainly), HORUS now has something of the following qualities:

- a large number of elements and relations in the RDB.
- a corresponding NDB that includes relations (serls) and their respective R-Set lists and is more comprehensive than the scope of relations in the RDB.
- an ongoing process of executing and evaluating inferences that belong to the different elements, including a realtime aspect of this execution, since some of the inferences may be time-related or event-driven, or event-sensitive (i.e., connected to events that may only exist through the exploration by HORUS of the so-called "outside world" (this is being explained more below).
- an ability for users (human and automaton) to pose different types of queries and tasks to HORUS for evaluation, using its knowledge base as the source for determining facts and relations; i.e., pertinent knowledge.
- an ability and program structure by which the system can modify different propositions and inferences and in general any of the relation information pertaining to one or more elements, in experimental and exploratory fashion, in order to innovate, to discover, and to invent new relations and in fact new elements.

There are several modes in which HORUS can be used and these will be described in succeeding sections now.

HORUS as an Active Analytic-Synthetic Thinking Brain

…

THE REST IS UNFINISHED !!!!!!!

(sorry, I only got this far tonight, and I still have to reconstruct a lot from memory, emory, mory, ory, since many notes were only handwritten in the famous Green Notebook, now long gone…)

OTHER STUFF

EDITING and MODS
The dynamic modifications can affect only a particular user or group of users. Results from examination of the different dynamical changes can and will result in modifications to the master-editing of the Syntopicon and its periodic revision by the editors.

Master-editing by the sysadmin team is always administered and finalized manually, no matter what tools are employed.

Examples

Def-Use	Op	Def-Use
2.3	4.3	15.0

2.3	5.6	14.0
3.4	20.1	14.0
3.4	13.7	21.5
8.7	3.2	7.0
8.7	17.5	25.5
12.0	6.6	30.5
12.0	9.9	28.0

In the database representation (not SQL format but something we have yet to work out) there is the following:

Element Def-Uses (DU)
Element Relations (REL) (always a triplet, but…there can be associated DUs and associated RELs)

So we have to have a way of keeping them together yet distinct

Without heavy computation (thus ruling out a lot of prime number methods, encryption algorithms, set theory, even lists).

Semantic (syntopical) communities

There is a hierarchy of semantic communities that define different users in the logical sense for HORUS.

Avatars (Personas, Personalities) (subset or aspect of an individual user - in the classical sense of a user having one userid+password defining one user logical space – having filters that can be applied to the Syntopicon usage and these create variations and dynamic qualities to the way the Syntopicon is used during the activation period of that filter).

---- the rest is more or less classical in the Unix sense of users, groups, etc. ----
Users (one or more avatars – there is always at least one, namely "self")

Groups (two or more users)

Supergroups (one group plus one or more group or user)

Megagroups (two or more supergroups)

Universe (everybody)

Syntopicon::universe is controlled by the system administrators and editors

Syntopicon::megagroup … Syntopicon::avatar(x) can evolve a variety of dynamic alterations that are the result of usage by the members of the "community."

Major Algorithms Employed Within HORUS

[these will be described later – mostly pull in text from prior papers and notes]

HAL

SOMA

SONON

Algorithms and Systems That Employ HORUS and its Syntopicon

Algorithms

METI
[future – fold in the material from those papers]

METI uses a special API that can work at a low-level directly with the NDB, known as the NDB_API. These are calls to the disk-based or diskless representation of the NDB for a particular HORUS instantiation.

Systems

UMA Collaborator
The UMA Collaborator software is an example of a C3DE – a Collaborative Cognitive Community Development Environment and it is basically a large-scale highly-interactive web portal with databases for contemt management and knowledge acquisition. The UMA Collaborator can be used in many different application contexts. It can also used HORUS in a modular design-in modality, or it can make use of a "universal" open and public HORUS that is stand-alone. The particulars depend upon the design intentions and requirements for the individual UMA application and UMA is designed to support access of HORUS as an internal, closed module or as an external web service.

In both cases the primary interface between UMA code and HORUS is through HTTP protocols for communication, using PHP and SQL functions as the primary coding method. The UMA code uses an API library known as HORUS_API that provides all the functions needed for bidirectional communications. This is described separately in the HORUS_Design documentation.

TransPrint Knowledge Base
The TransPrint Knowledge Base (TPKB) is a particular and very large information system designed for the publishing and content development (including post-production, post-sale dynamic web content) of printed books and internet sites associated with books (and vice versa). The TPKB is very much like a specialized implementation of HORUS is some respects but includes the HORUS software and utilities within its overall systematic structure along with other software that does not directly interface with the HORUS component.

This is a clear example of a modular application of HORUS as was mentioned earlier in the Introduction. The same interfaces using the HORUS_API library are employed.

In general, any system that obtains data or provides data into HORUS in any of its forms will use the same library. Internal to the functions in that library, there can be changes and upgrades, therefore, which do not force rewrites to the API and to the user applications.

[2] The term KIDILI has been introduced as an expansion or extension of KDI (knowledge discovery and inference), the latter being in the common vernacular of AI (artificial intelligence). KIDILI stands for: Knowledge Imagination, Discovery, Inference, Learning and Invention, these terms being understood and used here to form a logical and historical sequence for how people move from the point of getting some raw creative ideas and "fragments" to where eventually there is both learning and the invention or innovation of new concepts, relationships, objects.