CUBIT

Constructor Networks
for Understanding and Building
Intelligent Technology

Specifications and Technical Development Workbook

~~~~~~~~~

Design and Proto-Code

version 1.0.1; initiated 14.May.2018
(last edit mjd 1700::18may.18)
~~~~~~~~~

Contents

There is a large amount of material here left from the earlier architecture document,
CUBIT-knowledge-constructor-network_arch-design-wkbk_v1-0 (.doc, .pdf)
This is intentional – it is for reference, and I am using it progressively in developing the formal specs. This material is Unchanged from that document. All the new stuff is in the beginning sections.

Preface

These are notes, divided into sections. Gradually they will self-arrange into a logical system.

Then will come the formal specifications and design and all the rest.

Intro 1 - Intelligence is a process of quantum biological computation

Here is a set of assertions. This is the basis of the logic of CUBIT. These include some fundamental principles that guide the motivation, content, and computation involved in CUBIT.

Biology is a quantum physical process involving large networks of stochastic processes which evolve a common stable structure in the course of energy optimization. This leads to everything in the biological world including natural intelligent behaviors.

Brains are quantum computers, but of the non-Turing type, although in the process of neurological evolution, Turing-type computation becomes possible. First we imagine, then we conceive thoughts, and later we count, list, sort, and only still later we do deductive and inductive reasoning.

One way to pursue quantum computation (QC) is to think in terms of some device where everything is taking place – the computer, the processor. This is the approach of most research in so-called QC. There is a processor, with an array of qubits, the entanglement process is established physically between the qubits, and then the algorithm is executed, and as long as there is no decoherence of the entanglements, there will be a valid result from which to determine the best answer.

Another way is to think in terms of a large stochastic, random-like network, where each individual node is randomly configured and processing independently of all the others. The network nodes correspond to the array of qubits. There is no initial or pre-set entanglement between all the nodes of the network. Instead, something like entanglement evolves over time. How can this be detected and used?

??? ??? ???

In the vast randomized stochastic network, at some point, structures will appear. Like cellular automata, but not only local, but extending over the span of the network. Here is where resonances begin, and leafing into CQER.

This is what we must aim to understand better and to know how to detect and to control.

Intelligence requires randomness and also turbulence that emerges in the randomness. The turbulence is the seed of structure, and some of it simply dissipates back into the random sea, but some takes root and begins to crystallize. That is what leads to all sorts of cognitive behaviors.

Intro 2 - Cubits as Active Agents

Each cubit is an agent. It is an informational machine, an in-forming machine. Information is not only information as content but information as process. In terms of classical computer science, cubits are both data and program (instruction).

Information may be represented in many formats and media, but what matters is that it can be converted, transformed, in a manner that preserves the information losslessly. For now, we consider primarily cubits that have all their information in digital form (bits and bytes).

Cubits have content, like text, graphics, and tag-based (e.g., HTML-based) links to other objects, both cubits and otherwise (e.g., URLs to audio, video, other websites, etc.).

This is information, and it may be used in two ways ("voices" – akin to linguistic grammar) - passive or active, and this variety of use can take place at different times.

Passive information – it is what we traditionally think of as content, media – data, which does not act, and which can be changed by some program, some execution of instructions.

Active information – it is what we traditionally think of as program, as instruction set, as code - something that is "executable."

But all of this content is dynamic and can be changed over the course of time (usage). This includes information changing from active to passive and vice versa.

When there is interaction with a cubit, many things can happen.

Its content is accessed, made available, presented. What does this mean?
The user can see certain things (not necessarily all).

Every time a cubit is accessed, it remembers something about this.

We need to decide the level, the scope of detail.

But now we speak about the phenomena themselves.

A cubit has at some state *s* a configuration. What is a configuration?

The configuration is everything about the cubit geometry. Its structure – its faces, edges, vertices – its surface. And its volume, its interior. And its exterior, its containing-space. And its orientation within that space.

All of this information completely describes the cubit.

Mapped to all of the above – in ways that we can determine and decide – is information. It is content, substance, the atoms of the cubit.

This is information, and it may be passive or active, and at different times.
Passive information – it is what we traditionally think of as content, media – data, which does not act, which can be changed by some program, some execution of instructions.

The entire content of a cubit may be considered as data, but it may also be considered as a program, as code, and the cubit itself is a function, for which the rest of its data, and other information available to it, constitute the arguments, the parameters.

I. CUBIT Taxonomy

This is the basic classification schema for cubits. Note that this is Revised in this document and supersedes earlier architectural workbooks.

(cubit /* knowledge constructor component */

(**cubit-simplex** /* **cubit-σ** --- cubit-simplex; fundamental, unitary cubit logic element */

(monad /* elementary process unit – computational and (optional) physical components */

(**polymon** /* monad that contains information which can be interpreted as both (either/or, and/or) data and code (program, instruction) – this is the most complex type of monad.

Components: content (thema, preface, body) and meta-content (history, comment, use, geometry) in the format of an s-expression:

```
( thema
 (preface)
 (body)
 (meta-content) ) */
```

/* … there may be sub-classes within class "polymon" */

) /* polymon */

(**monomon** /* monad with rich data but no code, consisting of structured blocks of text (abstracts, quotes), graphics, audio-video media, URLs, and simplified, well-contained multimedia dynamics such as script-driven animations, all of which are contained within a cubit preface only; also known as a "cue" type of cubit */

Components: thema, preface-only (body = NIL, meta-content = NIL) in an s-expression format:

```
( thema
 (preface) ) */
```

/* … there may be sub-classes within class "monomon" */

) /* monomon, a.k.a. "cue" */

(**logon** /* monad with data or code in the form of an s-expression */

Components: (variable contents):

```
(s-expression) */
```

) /* logon */

) **/* monad */**

(binar /* elementary agent unit – (Biological-like Intelligent Neural Adaptive Recognizer) – computational and physical components, both data and code */

/* binars each contain one or more polymon-class monads in a network structure comprising additional data and code contained within one or more logons */

(**humint** /* a human-linked agent, a cubit associated with a human user who is operating through some type of interactive device such as a phone or computer */)

(**syint** /* a synthetic-intelligence agent, implemented in software and optionally in specialized hardware, and in theory designed and engineered by either a human or other syint(s) */

(**cybot** /* generic, limited-autonomy, limited-modifiability syint type of binar – computational element (analogous in some respects to "bots") */

(**cybernaut** /* a type of cybot dedicated to network-wide cybot management, coordination and control functions */)

(**cyberacq** /* cybot dedicated to data acquisition functions */)

(**cybercog** /* cybot dedicated to cognitive reasoning functions*/)

(**cybercom** /* cybot dedicated to communication functions*/)

(**cybersrv** /* cybot dedicated to servoactivation functions */)

) /* cybot */

(**mentat** /* self-organizing and semi-autonomous binar – computational element with greater complexity than a regular cybot (analogous in some respects to "bot masters" - subtypes include same classification order as cybots (family) */

(**m-cybernaut** /* a type of cybot dedicated to network-wide cybot management, coordination & control functions */)

(**m-cyberacq** /* cybot dedicated to data acquisition functions */)

(**m-cybercog** /* cybot dedicated to cognitive reasoning functions*/)

(**m-cybercom** /* cybot dedicated to communication functions*/)

(**m-cybersrv** /* cybot dedicated to servoactivation functions */)

) /* mentat */

(**kyborg** /* type of syint binar specifically dedicated to KYBEROS-type hyper-encryption and hyper-security tasks */)

(**soph** /* higher-complexity, higher-autonomy syint binar – essentially an assessor, evaluator, and coordinator of mentats and cybots and elemental binars – subtypes include the same classification order as with cybots and mentats */)

) /* syint*/

) /* binar */

) /* cubit-σ *aka* cubit-simplex */

(**cubit-complex** /* **cubit-κ** --- group of cubits operating as a sub-net */

(**binet** /* configuration of binars; a binar community, colony, organization; this may be distributed over a network of devices including servers, standard computers, mobile devices such as phones, tablets, IoT devices, etc. Note that this can include human user-agents and/or other types of agents including cybots, mentats, kyborgs and sophs*/)

(**monet** /* configuration of monads which may be any combination and number of polymons and/or monomons (a.k.a. cues) */)

(**luxnet** /* computational configuration (environment) for the control and operation of cubits in applications involving interactions and interfaces with conventional computing devices and humans; the configuration may be widely distributed as above, such as in a multiplicity of devices within several distinct networks. */)

) **/* cubit-κ *aka* cubit-complex */**

) **/* cubit */**

II. Basic CUBIT Specifications

Attributes and Behaviors

What is presented here is the general specification for a cubit in terms of its attributes (content – data/code, and meta-content – data), and its behaviors (processes, based upon those attributes). Note: Not all of this needs to be implemented early-stage or for all cubits, particularly the potentially data-rich meta-content. However, things are presented here for consideration, planning, and eventual implementation.

Attributes are properties, and they may be data or code or a mixture of both. Attributes define what a cubit is.

Behaviors are actions, processes, what a cubit does and can do. Behaviors are the result of certain attributes being processed. The information in the attributes specifies what and how the behaviors take place.

§ 1.0 Attributes

Cubits have two attributes: content and meta-content.

§ 1.1 *Content Attributes*: Digital information that is dynamic (modifiable, replaceable). It may be function both as data or code. It is considered to be material that is directly and consistently relevant to the function of the cubit.

There are three types of content: thema, preface, and body.

Thema: the basic (dominant) subject, theme, topic (this serves as its formal name in the cubit_namespace). Each cubit has one thema, although it may have an open-ended number of associated themas (see: "Relations," below under "Body content")

The thema also constitutes the function-name of the cubit, insofar as the cubit may be understood to be an expression (s-expression) that defines a function.

Preface: (a.k.a. "Cue") - passive content, structured, organized layout for access and presentation (display, interaction). There are four types for any cubit preface:

- **Precis** (abstract) – one short text, generally under 300 words, maximum 400 words, that gives the essence of the thema for this cubit.
- **Significators** – a list of objects, events, persons, other notables, including quotations, which individually and collectively comprise a strong, comprehensive perspective about the cubit thema.
- **Visuals** – a list of images and display configuration specs.
- **Links** – URLs that lead to webpages and especially to video and audio media.

Body: active and passive content ("body-content") that is created in the context of cubit use. This usage consists of interactions of two types:

- Self-initiated (controlled by code within the cubit)

- Agent-initiated (controlled by binars, either human agents or cybot agents)

Body-content may consist of any of the following types, arranged in lists (s-expressions), and there is no limit to the number or size of the elements of this content:

- **Plain text** - text without links or other programmatic functions.
- **Image** - any standard graphics.
- **Media** - other types of digital media, directly embedded in the cubit dataset.
- **Link** – a URL.
- **Relation** (r-expression, rexpr) - a special syntax for expressing relations of the three basic types (definition, association, inference) that exist among cubits. These relations may be of a wide variety and can directed to/with any other cubit. Relations have the list format: (rexpr [relation expression])
  R-expressions are a very important part of cubit development and will be closely linked with the different functions that cubits perform, and which will be expressed in the code components of the body-content. *(See notes on Relations, below)*
- **Code** - executable program source code, which may be in a variety of languages and syntaxes. This code, when executed, may perform a variety of actions pertinent to the cubit and its contents, and the environment in which the cubit is being used (displayed, interacted-with). Code may interact with any other elements of the cubit, both other content attributes, and meta-content attributes. Code has the list format: (code [language-identifier] [source code])

The specific syntax details for cubit content are still being refined – in particular, how to represent various media elements, texts, URLs, and other things that will come into play.

§ 1.2 *Meta-content Attributes*: Digital information that refers to the usage-history and inter-cubit relationships associated with the particular cubit. This is dynamic (modifiable, replaceable), and it is purely data and not code. It is considered to be material that is relevant to some functions of the cubit and to the activities of different agents that use the cubit.

There are four types of meta-content: history, comment, use-context, and geometry.

History: passive content consisting of entries that describe a meaningful event involving the cubit: An object of this type will have four possible elements (but not all will be necessary or appropriate for a given event and its history entry:

- Agent (user)
- Timestamp (for the action involving the cubit)
- Initial thema (where the user began in this iteration of working/navigating with cubits)
- Changes (list of pointers to cubit elements changed by this user) [tbd – what level of detail of tracking changes will be kept? Are their archived versions of cubits (probably) and for what details?]

Comment: some type of body-content that an agent adds to a cubit. This will contain:

- Agent (user)
- Timestamp (for the action involving the cubit)

- Comment (either the actual comment, if only a block of text, or else a pointer (digital pointer to the comment entity) if the comment is itself constructed as a cubit

Use-context: this is logical-proximity oriented body-content that will contain:

- Agent (user)
- Timestamp (for the action involving the cubit)
- Cubit-space (list of other cubits that have been active in the same general space of activity by the agent – in other words, what other cubits may been in the workspace of the user in dealing with this cubit)

Geometry: everything concerning the placement of cubit information within a visualization-space, and extensible also to optional 3D physical symbols of the cubit. Note that the Thema (name) and any meta-content information is always handled in a common manner for all cubits (e.g., for example, the thema may go always in a particular relative position and any cubit meta-content to be displayed (visualized) may go within a pop-up window that appears next to the visualization of the cubit):

- Attribute element (one of the Preface or Body attribute elements)
- Relative location (x, y, z coordinates, relative to a defined (0,0,0) for the cubit's geometrical representation)
- Geotype (what type of object to be used as the visual shape – e.g., cube, tetrahedron, octahedron, other polyhedra)

There is still a lot of development to be done regarding meta-content. It is not yet as stable as regular content.

These details will be progressively determined later in the design process. Initially, meta-content can be accessed in simpler ways separate from the stylized visualizations used for the primary content of the cubits.

§ 2.0 Behaviors

Cubits behaviors are of these basic types, all based upon functions that are activated within cubits:

Internal cubit functions

These involve code that exists within the cubit attribute set (in the "body" attributes).

External cubit functions

These involve code in other cubits that send messages to a given cubit and the processing of that message results in some action taken by the target (recipient) cubit. Internal cubit function(s) receive the message and perform some processing as a result.

Global system functions

These are not within specific cubits but have effects on one or more cubits (typically on some set), but this is accomplished also through message-passing that results in internal cubit function activation.

§ 3.0 Cubit Instances and Categoricals

What we have been describing thus far pertains to cubits as categoricals (classes, general types).

There is only one categorical for any cubit. The cubit thema is a unique identifier. Categorical cubits may be created by humints or syints (but initially, they are created only by humints, namely, the designers and builders of CUBIT).

Cubits can also have instances and there is no limit to the number and variety of cubit instantiation.

A binary (humint or syint) can generate a cubit instance. It is a “local copy” of the categorical cubit, inheriting all its attributes at the time of its creation (There can be some option for enabling the creator of the cubit instance to limit the attributes that are “instantiated” but technically, this is a change-operation, something post-creation. Thereafter, its content will vary in accordance with its use by the particular binary (agent) that created it.

Can cubit instances be shared or transferred among binars? Why not – it seems like a good feature to allow.

§ 4.0 Cubit s-expression (sexpr)

Here is a sample of how a cubit is represented formally, followed by a partial example.

[Basic s-expression]

```
(thema
  (pref
    (prec text-string)

    (sign text-string1 text-string2 text-string3 ... )

    (vis <img src="http://server1/image1.jpg">
         <img src="http://server2/image2.jpg">
         <img src="http://server1/image3.jpg"> ... )

    (link http://server1/page1 http:// server2/page2 ... ))   /* precis */

  (body
    (text text-string) ...
    (image image-data) ...
    (media media-data) ...
    (link URL) ...
    (rexpr relation-expression) ...
    (code python code-block) ... )   /* body */


  (hist
    (agent timestamp starter-thema some-change) ... )


  (comm
    (agent timestamp comment-text) ... )


  (use
    (agent timestamp (cubit-list)) ... )

  (geom
    (preface (x-offset y-offset z-offset) polyhedron-form) ... ) )

/* cubit */
```

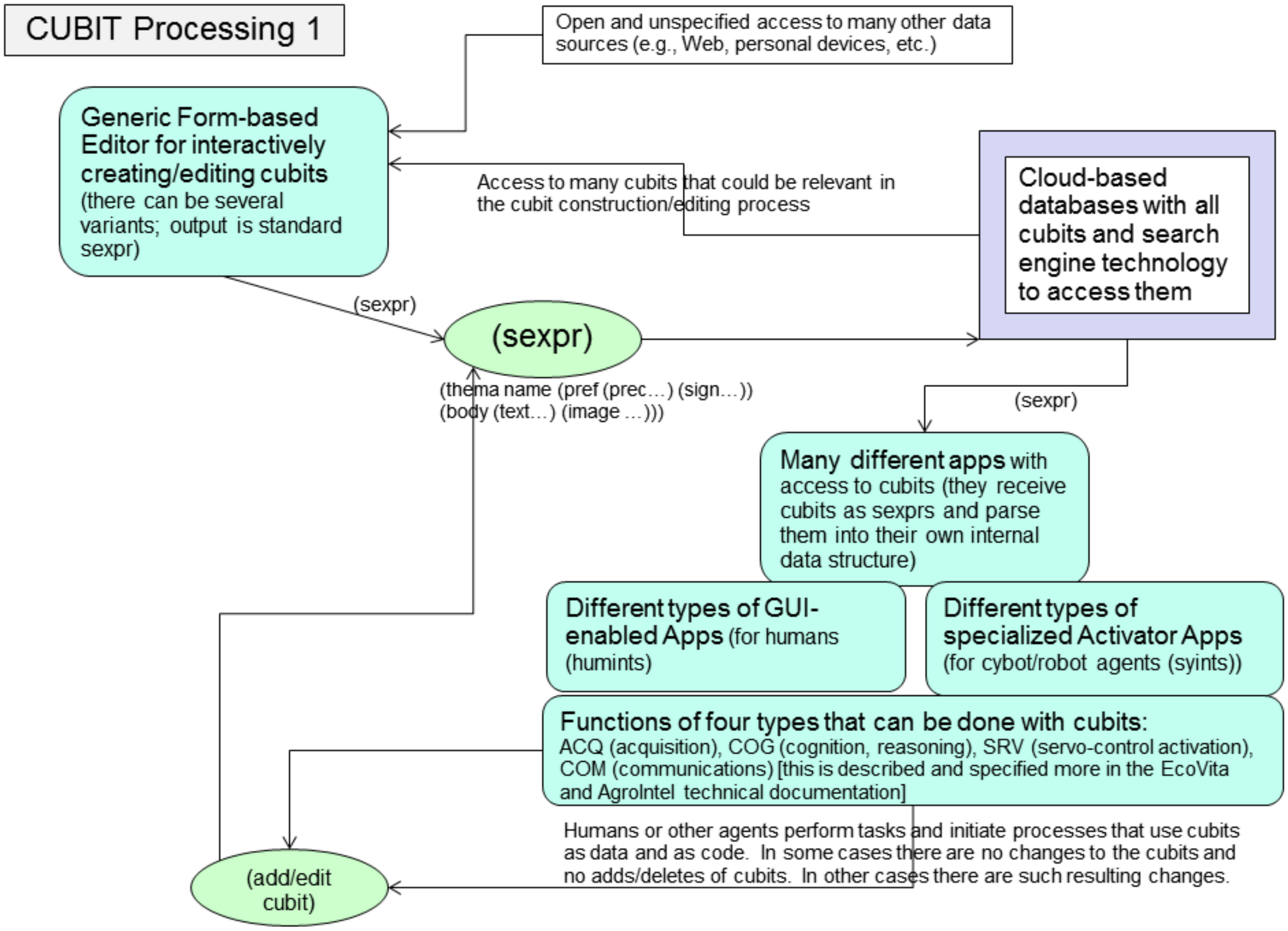
[An Example]

```
(quantum-mechanics
  (pref
    (prec “For nearly a century since the famous Solvang Conference and the
famous debate between Einstein and Bohr, there has been as much uncertainty
and indeterminacy about Quantum Mechanics and everything “quantum” - in
physics and within the whole of society, as there is in Heisenberg's famous
Uncertainty Principle.  Metaphors abound with cats who are simultaneously
alive and dead…”)

    (sign “[The] atoms or elementary particles themselves are not real;
they form a world of potentialities or possibilities rather than one of
things or facts. -- Werner Heisenberg”
 “Quantum physics reveals a basic oneness of the universe. - Erwin
Schrödinger”
““But why then had Born not told me of this 'pilot wave'? If only to point
out what was wrong with it? Why did von Neumann not consider it? … - John
S. Bell, On the impossible pilot wave. Foundations of Physics 12 (1982)”
     ... )

    (vis <img src="http://server1/image1.jpg">
         <img src="http://server2/image2.jpg">
         <img src="http://server1/image3.jpg"> ... )

    (link http://server1/page1 http:// server2/page2 ... ))   /* precis */

  (body
    (text “Additional text including HTML”) ...
    (image additional-image-data) ...
    (media other-media-data) ...
    (link some-URL) ...
    (rexpr (SRO “quantum potential” model “double-slit experiment”)) ...
    (code python some-code-block) ... )    /* body */


  (hist
    (something) ... )


  (comm
    (something) ... )


  (use
    (something) ... )

  (geom
    (something) ... ) )    /* cubit */
```

[Here, later, will be an EBNF formal syntax diagram for a cubit]

§ 5.0 Some Figures and Diagrams

Go to the next pages →



CUBIT Processing 1

Open and unspecified access to many other data sources (e.g., Web, personal devices, etc.)

Generic Form-based Editor for interactively creating/editing cubits (there can be several variants; output is standard sexpr)

Access to many cubits that could be relevant in the cubit construction/editing process

Cloud-based databases with all cubits and search engine technology to access them

(sexpr)

(sexpr)

(thema name (pref (prec…) (sign…)) (body (text…) (image …)))

(sexpr)

Many different apps with access to cubits (they receive cubits as sexprs and parse them into their own internal data structure)

Different types of GUI-enabled Apps (for humans (humints)

Different types of specialized Activator Apps (for cybot/robot agents (syints))

Functions of four types that can be done with cubits: ACQ (acquisition), COG (cognition, reasoning), SRV (servo-control activation), COM (communications) [this is described and specified more in the EcoVita and AgroIntel technical documentation]

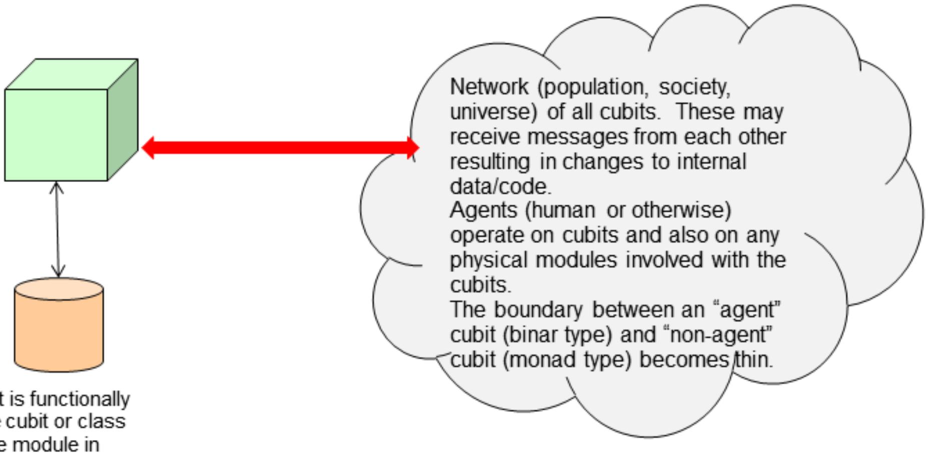
Humans or other agents perform tasks and initiate processes that use cubits as data and as code. In some cases there are no changes to the cubits and no adds/deletes of cubits. In other cases there are such resulting changes.

(add/edit cubit)

CUBIT Processing 2

Open and unspecified access to many other data sources (e.g., Web, personal devices, etc.)

Cubit with content (preface, body) and meta-content. Body includes optional executable code for various control functions (ACQ, COG, SRV, COM). Some of this will pertain to any associated physical cubit-module(s) linked with the cubit.



Network (population, society, universe) of all cubits. These may receive messages from each other resulting in changes to internal data/code.
Agents (human or otherwise) operate on cubits and also on any physical modules involved with the cubits.
The boundary between an “agent” cubit (binar type) and “non-agent” cubit (monad type) becomes thin.

Cubit-module – a physical object that is functionally and/or symbolically linked with some cubit or class of cubits. (Example: an S-Water type module in the AgroIntel system, linked with one or more cubits that contain information and functions relating to agriculture, irrigation, energy and water conservation, horticulture, farming, etc. Informationally and computationally, this module is treated as data and code within the cubit as a subset of its attributes.

Principal uses of cubits:			
Communication	Education	Research	Production
Examples:			
Artists + Scientists	Train children and workers	Basic/applied	Smart Farming

III. Notes (New, and also from earlier Architecture Workbook)

These extend into other Sections which have been included from the 14.May (12:25am) Architecture document,

CUBIT-knowledge-constructor-network_arch-design-wkbk_v1-0 (.doc, .pdf)

Using Cubits for Understanding How People Think, Reason, and Solve Problems

[Coming later… We want to show here how the use of cubits, including the effective use of randomness, stochastic behaviors, and anomaly detection – including patterns of anomalies – is very important for learning, for the substrate-development of intelligence and problem-solving, in individuals and among groups. This includes the importance of trial-and-error – the essential importance of making mistakes!

Among other things, we will make a major distinction from what is dominant in current AI and social network circles, as typified by the Google "Selfish Ledger" model and Google Duplex, and a lot of other "mistakes" being performed by mostly internet companies and certain academic followers of the "total information capture and mastery" philosophy.]

Examples:
For now, see the proto-cues that are in cues-master-workbook-vol-01-1.ppt, particularly the first, on the general thema of quantum mechanics.

Cubits are bots, and not just bots

Cubits are like bots in botnets, in another way of thinking, only that there is no singular botnet commander. The network itself as a whole is what controls all the agents (bots). That control is both explicit and implicit, and it ranges from "influence" to direct control.

All of this information can be represented in a manner similar to how documents are processed using HTTP, namely, by XML and other data serialization (e.g., JSON). However, cubits will be implemented primarily using s-expressions, because that format is simpler and more versatile from the standpoint of representing executable code as well as passive data such as in static documents.

Beyond Blockchain

Blockchain has extreme limits by virtue of precisely what it does. It have have value in certain areas but not as a universal method for tracking and validating events or “transactions.”

What cubits do that is different and more universally useful:
Note that in what it being described here, there is a built-in mechanism for detecting anomalies and disturbances, and this can be employed to detect any tampering with cubit histories, and this does not require anything of the computational overload used in blockchain..

Cubits as Complex Geometrical Propositions

(Some of this is already reflected in specs above, earlier – about attributes and behaviors)

Each cubit is a geometrical object. It is some type of abstract polyhedron.

The geometry reflects the use and the meaning of the content-elements in the cubit.

What is this contents?
Assertions, propositions. They are Relations which are either definitions, associations, or inferences.

Definitions simply assert, declare, state something, unitary.
Associations assert a connection between a cubit to others – one or more.
Inferences assert more complex relations.

As such it consists of these elements:
Faces, Edges, Vertices.
It also has an interior, a volume, and an exterior, a containing-space, the space in which it exists, moves, acts.

Let's say that we begin with the regular polyhedral, of which there are five.

Each cubit is like a special kind of HTML document. It has additional markups that point to different "atomic" objects. These are the things that go into a cubit.

Basic-content-data
Texts – this includes short paragraphs, quotes, all the text elements designed into cues.
Graphics – images, of whatever type
URLs that go to something else
Visualization/animation scripts that do things to the cubit as it is being viewed, used, manipulated

Basic-relations
Definitions – this level can include keywords (tags)
Associations
Inferences

Meta-data
Connections made to other cubits
Who made the connections, when, where, etc.

References to this cubit by other cubits

Primary (Initial) Cubits and the Primary Cues

(this is from the earlier architecture document, mostly unchanged)

Cubits exist and can exist on many topics, subjects, areas of focus. As CUBIT grows and evolves, there will be an increasing population of many cubit types, especially polymons (within the monad class) and cybots (within the binary class). However, at the outset of system deployment, we begin with a Primary Set of cubits, also known as the Primary Cues. These are cues, not complex polymons, and they do not have agent capabilities. As everything evolves, cues can evolve into polymons, and into binars.

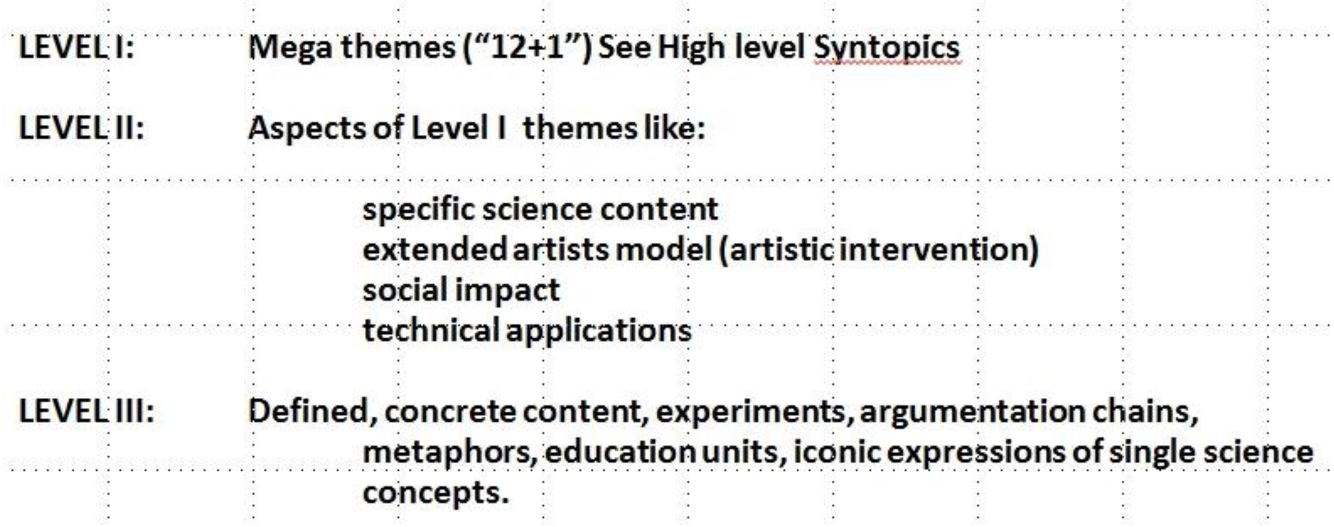
The initial or primary set of cues have been chosen for a variety of reasons pertaining to prior knowledge, perceived needs, expected reception, and projected support. From these the expanse of all other possible cubits opens up widely and with potentially unconstrained diversification.

The primary cues are the first ones being designed and implemented for distribution. They also provide the initial basis for knowledge building-block objects. These are not listed here in order or priority or implementation, but in relationship of ontology, scale, and causal connections.

[1] Quantum Information and Transmission (QIT)
[2] Quantum Computing (QC)
[3] Nanomedicine (NanoMed)
[4] Climate and Environment Dynamics (ECO)
[5] Human-Robotic Interactions (HRI)
[6] Internet of Things (IoT)
[7] Smart Farming (AgroIntel)
[8] Smart Energy (IntelErgy)
[9] New Tech to create and not eliminate jobs and careers (JOBS)
[10] Social Impact-Influence-Control Tech (SOCTEC)

There are three general types of topical focus or subject matter among cubits. These applies to both monads and binars, but it is principally relevant to monads. These types define the main areas of interest or attention for the cubit entity. There are Level I ("mega-theme") topics, also called themas or *Syntopics*. There are also Levels II (specific scientific or artistic content, social impact, technical applications) and III (well-defined, concrete content, experiments, argumentation chains, metaphors, education units, iconic expressions of singular science concepts). Initially, we will produce ten Cues that are Level III types.

Figure 3 below provides a concise summary of the three levels of cubit focus of interest.



LEVEL I: Mega themes ("12+1") See High level Syntopics

LEVEL II: Aspects of Level I themes like:

specific science content
extended artists model (artistic intervention)
social impact
technical applications

LEVEL III: Defined, concrete content, experiments, argumentation chains, metaphors, education units, iconic expressions of single science concepts.

Figure 3 – Three levels of cubit topical focus

Our choice of the Primary Cues that we introduce to the world is very important. They must be attractive, magnetic, and with diverse appeal to many markets. They must become brands that can be easily recognized, identified, and served as "magnets" for people and organizations. One tentative set of ten (10) such initial (introductory) Level III Primary Cues may consist of the following (numbered references connect with the above list of ten Level I themas):

New metaphors, symbols and illustrations for quantum physics, particularly superposition and entanglement ("Schrödinger's Cat finally escapes the cage") [1, 2]

How "big data" and "data mining" invisibly shape social thinking, expectations and market behaviors [10]

Quantum computing as it is, as it is promised, and as it may become in the future [2]

Quantum information transmission, signaling, and privacy [1]

Smart Farming 101 – intelligent control and optimization in home gardening and in the commercial small farm [7]

Creative approaches to designing new icons and symbols for counter-intuitive concepts [1, 2, 5, 10]

Artists and scientists interacting and collaborating – differences and complementarities [1, 2, 5, 9, 10]

Practical applications and uses of "internet-of-things" for food, energy and health [3, 4, 6, 7, 8]

Smart Energy 101 – hybrid energy production for the home, backyard, and in urban and rural environments [8]

Robots, home life, jobs – the realistic technology, the limits, the hype, and the social effects and how to plan intelligently for the future, for families and communities [5, 9]

Cubit Functions

[conceptual, theoretical] (also from the earlier architecture document)

Cubits are functional entities in the formal sense. They do something and the result is a transformation. This action, this transformation, is something highly interactive between the user, the person or group using the cubit and its content, and all of that content including both digital and non-digital media and mechanisms.

Cubits perform multiple types of functions. These functions are non-exclusive and fully cooperative. Cubits can be viewed from the perspective of all these different functions that they can perform.

There are four global function types:

Utilitarian – the function is for accomplishing some intended goal, some outcome, with respect to the ways that the knowledge can be transmitted and used.

Operational – the function performs some transformative operation. Something changes (presumably in the mind and behavior of the user(s), something is gained, clarified, perfected.

Purposive – the function serves the goal, the objective, to increase knowledge about something on a more theoretical level, or on a more practical and application-oriented level.

Modeling – the function provides a way to model, including to predict and forecast, the behavior of some system, some STEM object of interest.

These four different function types are described here:

Utilitarian Functions

Cubits are designed for use in multiple capacities and purposes. There are five general utilitarian functions – these define how and why cubits are designed and employed:

Media Distributions – documents and works of art (not restricted to text documents) that are used typically for multiplicative, distributed, far-reaching effects through analog and digital media. Examples include: short stories, comics, graphic novels, conventional novels, manifestos, whitepapers, formal research papers and reports, audio and video creations, and other art forms).

Briefings to initiate and facilitate interaction and constructive dialog among artists and scientists as well as persons in the financial, management and socioeconomic sectors. The focus here is upon the initiation and facilitation of dialog between diverse persons and groups, spanning all societal segments, such that the STEM content, along with other information and knowledge, can be useful to the requirements and tasks of such participants. The ultimate purposes may be STEM-related or may be in diverse socioeconomic matters such as employment, immigration, and the overcoming of barriers and conflicts among different elements of society.

Science Communications (SciCom) – typically in the context of formal STEM-focused academic presentations (symposia, congresses, seminars, courses, open public access) and individual/group use (e.g., online resources and physical kits). There is some STEM topic, somebody wants to communicate their work and/or the general field more in ways that will be more cohesive, intelligent, attractive, understandable, clear, cogent, and without hyperbole, exaggeration or pseudo-science interpretations. This is what SciCom is all about.

Education Tools – for use in schools and/or as mobile gaming units, oriented to both children and adults. These comprise both computer-based tools and physical components (similar in concept to constructor kits like Knex and Lego), in both cases placing a strong emphasis upon 3D and 4D visualization techniques for learning and experimenting with concepts and mechanisms involved in the subject matters being explored.

Research Tools – for use in all aspects and levels of STEM research and development, from speculative exploration to proof-of-concept demonstration to prototyping and beyond to implementation of experiments and commercial-application derivatives. These cubit functions are among the more complex, detailed and specialized, since here in areas of direct and deep research there may be more knowledge elements and relationships that go into a particular cubit or set of cubits. However, these research functions are also available and accessible to the other functions and to all users.

Clearly, different objects, whether digital and online, or analog and physical, will operate according to multiple functions and will serve the interests of multiple user-audience types. Every cubit will be usable in some way for all five utilitarian function types.

Operational Functions

Cubits perform five general operational functions.

Cubits provide an alphabet, a grammar, an algebra and geometry for knowledge acquisition, learning, and innovative design. All of this is open-ended and flexible, designed to initiate and sustain creativity and invention, and making use of form and structure in a manner comparable to the processes within architecture and engineering. There are fundamental logics and truths that are the guideposts and pathways for learning about different topics and how to employ the facts, models, theories, and speculative concepts intelligently. These translate into five operational functions.

Declarative-indicative-explanatory – what-it-is, what something is about, the facts, the knowns, an effective set of descriptive answers and resources for using the knowledge. A cubit enables deeper understanding of “what is” in some subject area.

Assessment-evaluative – ideas about the value, strength, utility of this knowledge, how it can be used, how it is validated. A cubit enables better discernment, choice and value-judgment by the user.

Interrogative-speculative – better understanding about what could be, what may be, what might be, with respect to some topics, some aspects, and this can also provide pointers and guideposts for how the user can go further in study and research.

Predictive-forecasting – ideas about what should be expected, with varying degrees of confidence and certainty.

Imperative – ideas about what should be done, what avenues and paths should be explored, investigated, performed.

Purposive Functions

Cubits perform two purposive functions, corresponding to the principle purpose or goal of some user – to understand how something is and works, or to apply knowledge and build something.

Theory (Concept)
Substantive knowledge about, the "what-is" information, how something works abstractly, generally, conceptually. Examples: what is inside a seed, a plant, an atom, a galaxy, what makes up such things, and how they work – what are the processes, the mechanisms, the dynamics. Examples from chemistry: how hydrocarbons and nitrous oxides exist, how they interact, what effects they have in the environment and upon biological organisms.

Praxis (Application)
How to build something, the technology, the methods, the protocols. Examples: a telescope, a hydroponic garden, a vertical airfoil wind turbine, a robot for agricultural operations, a robot for manipulating asteroids in deep space. Examples from chemistry: how to accurately measure hydrocarbons and nitrous oxides in air, water, solids, how to control them from an ecological management perspective, how to reduce their toxicity and their overall presence in the environment.

Modeling Functions

Cubits perform four general modeling functions.

Analytic – this is how something works, what goes on, what is the internal mechanism.

Synthetic – this is how something could be made to work, if various changes and alterations were to be made. This is more about a process that is more classical in its behavior.

Predictive – this is how something will behave if various parameters are changed. This is more about forecasting, and the behaviors of systems with non-linear outcomes.

Hypothetic – this is about more conjectural, speculative, radical outcomes based upon correspondingly more radical modifications of underlying principles, parameters, rules, axioms.

Cubit Logic 1

[conceptual, theoretical] (also from the earlier architecture document)

Everything within each and every cubit is based upon a simple logical structure. There are objects or entities and they are described in terms of the classes and sub-classes to which they belong, and they have various attributes or properties. All of these attributes can be connected in various ways with Relations.

Relations are of three elementary types:

Definitions – assertions, statements, fundamental propositions about the "ding an sich."
Associations – a relation expression involving two or more objects in the system.
Inferences – a relation expressing a rule which can be expressed as some form of basic conditional expression ("If x then y").

An excerpt here from a 2002-2003 whitepaper on the Syntopicon and its use in a system called the InCyclopedia (interactive internet encyclopedia and modeled upon the Britannica "Great Books" series) may be useful. Other technical materials are available including proto-software (but everything from back then needs to be properly rewritten).

> 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.[1] 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.[2]

RELATIONS

The following material originates in from HORUS (Hierarchically Organized Reasoning and Understanding System).

Each category of relation has three (3) varieties.

Relation	Subtype	Description

[1] Wittgenstein, Ludwig, "Philosophical Investigations"

[2] As a cellular automata (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.

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 creation is done only by humans using an interface that is similar to what is accessible in the HORUS system.

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 numeric matrix database (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,[i] 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.

Cubits are Informationally Dynamic

Everything in the cubit is *dynamic* information. This means, there can be many things that change over time. New references, cross-references, links, and there can also be true hypermedia design, in the original sense of hypertext as developed by one of the true "founding-fathers" of the web, Ted Nelson, with Xanadu and other developments. [3]

There are some very strong and valuable benefits of our approach with cubits, in all manner of media (not only text!). This includes the potential for full cross-referencing and two-way linking that is possible, between not only active (content-producing, content-editing) contributors but also passive users (reading, viewing, doing things repeatedly but without changing any content, for instance). People can see what others think and feel about certain concepts, images, texts, formulae, equations, models, and anything else.

Moreover, in this dynamism, there are be allowable, intentional, places for incomplete information, for things not-known, still-to-be-discovered. This is generally and severely absent in most textbooks, manuals, guidebooks, lessons, lectures, courses, and other media, across all five of the utilitarian functional sectors that cubits address. We can openly allow and serve this capability which in fact is essential for the learning and discovery and invention processes.

See Appendix 1 for more on the work long preceding Berners-Lee and others, including the whole "Google" episode in which we are now immersed. This is something important

[3] "A core technical difference between a Nelsonian network and what we have become familiar with online is that [Nelson's] network links were two-way instead of one-way. In a network with two-way links, each node knows what other nodes are linked to it. ... Two-way linking would preserve context. It's a small simple change in how online information should be stored that couldn't have vaster implications for culture and the economy." [Jaron Lanier, Who Owns the Future, New York: Simon & Schuster, 2013. p. 227]

for cubits, it makes a cubit, especially a "Cue" type of cubit, much more powerful than any static, passive document or piece of media.

Artists have recognized these points decades (and arguable, centuries) ago, and especially in the 20th century with the emergence of Dada, Cubism, and post-Cubist abstract art. But the STEM world has been slower, and this is a matter of psychology and neurophysiology for the most part. STEM has been dominated by the analytical, the discursive, the dialectical, the discrete, and now the digital.

But the creative, inventive, and expansively productive aspects of what STEM can be for the future of a naturally wholly-connected, wholly-interconnected society and ecosystem (! by virtue of macroscopic quantum entanglement-unity principles, for instance !) – this all requires a more wholistic, complete, comprehensive Mind, and that is where cues and cubits come in with a new way of seeing how people think and act with the knowledge that is given to them.

Cubit Logic 2

Any type of Relation can be added, extended, by the appropriate editor/monitor – whomever is given the permission to change things in the media.

Attributes can be added. Visualization software scripts can be added, resulting in things like clicking or automatically being led to seeing some demo, some model, or being in an interactive mode with some 2D or 3D simulation.

We want to enable cubits (cues) to have the ability to be increased, grown, matured, through their repeated use by different users.

This is why and wherefore that the true complete bidirectional hypermedia concepts and frameworks (e.g., Ted Nelson and Xanadu) become conceptually so important.

The creative, inspirational, inventive, innovative process that goes on in STEAM – in all the arts and sciences – is something that involves both conscious and unconscious thinking, and a lot of "sudden, out of the blue" thoughts, feelings, sensations, intuitions. This is non-linear thinking, but remember that linearity is really a subset of the "non-linear" – just like nice-and-simple Euclidean geometry is a subset of a more complete and open geometry that includes positive and negative curved spaces, both the Riemannian and Lobachevskii approaches!

And when a person (or collective group, working together) is interacting with a cubit, there is a lot of this non-linearity going on inside the brain(s) and interacting between the "observer" ("user/doer") and the "artspace" of the cubit!

So, we want to be able to somehow capture, record, collect, all of the meaningful and relevant interactions, all the "Aha!" and "Oooh!" and "What if?" and "How about?" types of experiences and thoughts, including so much that we do in our minds with internal visualization, internal geometrization, internal space-building and dot-connecting – and we want to not merely collect and archive that information but make it intelligibly usable to others.

This is how other users – and we ourselves – can better understand the thought processes going on, and through that, the explorations, scoutings, speculations, hypothesizations, and logic-building, that goes on.

ALL OF THIS IS OF COURSE VERY VERY VALUABLE COMMERCIALLY. And certainly there are people in different organizations, especially some major multi-national multi-industry corporations and holding companies, who will recognize this as soon as we meet with them and begin describing what we are doing.

We do not need to have everything "finished" in any sense of technology or media content. We just need to meet face-to-face with the Right Persons.

And personally, I believe that those Right Persons are presently accessible, and they are mostly living and working in cities contained within the "ring" that includes Paris, Brussels, Berlin, Moscow, Rome. And Vienna is basically at the centre, and in my opinion, in more subtle ways than are evident in the "general and conventional understanding of things." (Some of my sentiment comes from simply hours and hours of walking on many streets and being in many places, and both Marianna and I share some of these thoughts.)

Cubits as Knowledge Constructor Sets & Collections

[conceptual, theoretical] (also from the earlier architecture document)

This material will be developed later. It will be about how they work together digitally and physically. It will be easier to explain through graphics and actual physical "toy" models.

In brief, here is a preview:

On the computer, and with actual physical objects that literally represent – iconically, symbolically – different blocks of knowledge – people can "play" and assemble, disassemble, arrange, rearrange things – the cubits – and the software in their apps, whether they are using only mobile phones or larger laptop/PC machines – will enable them to see new relations, new facts and chunks of knowledge and especially techniques, styles, methodologies – presented to them.

There will be a lot of "why not try this?" suggestions that can be introduced and enabled. Some of these can include visual prompts, such as the nudging, vibrating, and other moving of graphical objects.

With physical cubits, these may indeed have some internal electronics, very simple, but interfaceable through Bluetooth, NRFID (near-field RFID), etc. imagine that as you move or make contact between two faces (sides) of some cubit objects, suddenly the phone or PC screen displays all sorts of interesting "potentially correlatable" information for the user to explore and review.

The ASI ("AI") components will help in making creative suggestions.

But a lot of all this can be actually done with very simple, straightforward common-sense content management. We are **not** talking here about some big software research task. Much of what we need is already available – fully open-source – and mostly in the simple management of HTML and various media.

But what needs to be remembered is that some person or persons is/are manipulating some objects on a screen, or in their hands. As they do certain physical, geometry-changing actions, which could be summarized loosely as:

- Contacting surfaces (faces, vertices, edges)
- Arranging objects in a pattern together, but not necessarily making direct contacts
- Drawing connections by making lines, arrows, loops, or simple "finger-sweep" motions
- Doing other actions that are comparable to some of the basic drawing steps in common application programs like Powerpoint, Photoshop, Visio
- Opening an object (e.g., a cube or dodecahedron) and graphically extracting some knowledge-objects – some of which may be other geometries such as discs, cubes, etc.)
- Putting one object inside another or pulling an object out of another
- Etc.

they are performing Knowledge Engineering Actions. They are performing some type of information extraction, combination, some type of logical operation, some type of Algorithmic Process. And in these actions, they are discovering relations, or being led and hinted-toward discoveries of new relations.

It's all about graphical, visual, tactile (or virtual-tactile) manipulations in a 2D, 3D or 4D space, where the points, the coordinates, the planes and spaces, are all associable with different pieces of information, with different knowledge relations.

What we are talking about here is the inspirational fueling and firing-up of learning methods and inventive processes within the minds of the people – as individuals or as a working group, through the use of 2D and 3D and fully-hypermedia objects that are either solely on computer/phone screens or also as special objects that are electronically-enabled, icon-like, sculpture-like, and often very artistically designed and styled.

Now, remember, that this is not only about doing “hard-science” STEM (physic s, chemistry, biology) but it is all open for and supportive of the psychological, social, and economic sciences and technologies. So I am making the bold but rock-solid, defensible claim here that with cubits we have the means to strongly change the ways that people perceive, learn, remember, and reason about psychology, economics, sociology, and of yes, politics.

Cubit Interactivity and the Art of Knowledge Construction

[technical] (also from the earlier architecture document)

What is described here is general, applicable to any cubits. We will begin with Cues, naturally. We also begin with things in the purely digital, computer-based world. Physical cubits will probably commence as non-electronic, non-digital objects, more like sculptures, or literally, as building blocks; progressively things will move toward their internal activity through the inclusion of electronic components.

A cubit has several types of information. Text, graphics, video, audio. Some are given as links. So the information is both superficial, on-the-surface, so to speak, and deeper, requiring some action to get to it.

There are multiple agents at work with cubits. One is the human user (who may be plural, in the case of groups of persons working together on something). Then there are the cybots or binars, the SI (synthetic intelligence) agents.

Any agent, human or cybot, can do virtually any and the same actions with cubits.

One action is to assemble or combine cubits, to use them as building blocks. What does this mean? It means that one is making a connection from one facet or component of knowledge in a cubit a, with one or more facets in another cubit or multiple cubits.

This can be done intentionally, voluntarily, by the user, or it can be through assisted by the cybot agent.

Think about combining tetrahedra, cubes, rectangular and other prisms, octahedra, dodecahedra, icosohedra, even rhombicuboctahedra.

Example Behavior of Cubits in Use

[technical] (also from the earlier architecture document)

A user (humint) brings up a cue. It is something about quantum physics, perhaps dealing with the topic of superposition and entanglement. The user has turned on all possible options for assistance, suggestions, redirection, and also for access to the activities and comments of others. Any of these data-rich features can be turned on/off at any time.

This is all implemented in standard HTML but with control of what is shown and accessible being handled both server-side and browser-side.

Everything the user does, unless he or she explicitly blocks the recording of transits, shifts, and other activities, is recorded. Thus the system – the community of relevant agents, some of which are linked closely with content cubits themselves, for others to use perhaps – has potentially all activity knowledge pertaining to any binary with any monad or other binary.

After some exploration within the cue itself, and perhaps to some external URLs, a cybot agent makes suggestions.

These may include examination of and interaction with other cubits. Some of those may be polymons, some may be other cues, some may be binars of different agent-types.

Suggestions can be presented in the form of visual links and "drawing together" motions between specific cubits or networked groupings (clusters or more dispersed networks) of cubits. Users can follow or act graphically, geometrically, as they please, to move in different directions. The original (actual) paths are always maintained in the vast recording mechanism employed within CUBIT for all interactions.

A user can see textual or graphical representations of what she is doing, what he has traversed, visited, commented upon. Everything is available in a bidirectional manner. You can see what X has done with cubit A, or you can see different instances of how cubits A, D and J have been brought together in some method by other users.

The simple way of thinking about all this is that there can be – Can Be – maintained an accurate record of every step-sequence by every agent, human or otherwise, in doing any activity with any cubit. This can be constrained and controlled and indeed must be do so, in order to refine all those activities to fit with what the current user/agent(s) are trying to do at this time. But out of the vast wealth of cubit activities, the system, through different assigned agents, can begin to learn more and more about patterns of activity, particularly patterns of interest and very significantly, patterns of reasoning – namely, how individuals and groups approach problems, look for solutions, search for ideas, and make associations. This helps to enable CUBIT to progressively understand, in a formalized, structured way, how agents Think. How people Think about le3arning something and solving some problems.

Gradually, CUBIT gains more and more knowledge about the world, yes, through the creation of more and more cubits. But in this process of being used, CUBIT gains more and more knowledge about how people (and its own agents) think about finding answers, learning, solving problems, and building new, creative, inventive structures from the knowledge that is at their disposal.

So, it can be said that CUBIT gradually becomes a system with more and more knowledge about how its users think and do knowledge engineering. This becomes progressively powerful because it can help users to make more shortcuts, more imaginative jumps, more analyses, more syntheses, more inferences, and in the process, cutting back unnecessary and unfruitful explorations. This is very valuable.

Cubit Dynamic Functions

[technical] (also from the earlier architecture document)

A cubit is a function. It may have all sorts of content that is passive in the sense of evaluating constantly to the same content, without change, and without effect on anything outside the cubit, but the abstract cubit is fundamentally a function.

In a manner of speaking, each cubit is constantly applied to some set of variables. What are those variables? It depends upon the state of the whole system, and upon the individual cubit and its local networks in which it may belong, temporarily or permanently.

There are algebras that can be designed for cubits. A rich history of work in process algebra from C. A. R. Hoare, Tony Hey and others can be applied to cubit processing, and this will be beneficial for building the intelligence factors into cubits.

Consider that a cubit is constantly evaluating itself, looking at all changes, all actions that have occurred or might occur.

What are the things that matter in cubit processing?

Has explicit_content changed? This is the type of content found in cues, for instance – actual texts, graphics, links. Explicit_content also includes attributes such as the geometrical shapes, colors, and other structural factors involved in cubit representation and display.

Has implicit_content changed? (We need to figure out precisely what this means, but I think it is content that is somewhere else but just not overtly, openly, in the cubit-as-displayed-and-used. Or does implicit_content also include the cubit logics?

Have logics changed? What are the logics of a cubit? These include certainly Relations - definitions, associations, inferences. But there may be other forms of logic. Functions that get performed under certain circumstances.

Have actions been done to or about this cubit by other cubits, particularly agents?
For instance:

- simple visits, contacts, by different agents (humint or cybot, etc.)
- comments made by different users
- links (arcs) drawn to/(activated) from this cubit to others, or links broken (removed, disabled, deactivated)
- rearrangements or reassignments of the cubit in various sub-nets

Based upon the parameter settings for the cubit and for the (enveloping, encompassing, inclusive) system, some of these actions will be ignored and others will be acted upon, resulting in some type of programmatic evaluation. The results can be:
limited to the cubit, internal
effects that are shared with others according to various rules
universally shared among all cubits

How is all this implemented, computationally?

Each cubit has some processing logic. It may be very modest, simple, and it could even be something like Javascript. There are seeds, kernels of code that can be attached to any cubit.

Some logic will be server-side, some client-side. Some will be distributed throughout a network of cubits and the processing machines involved in that aspect or domain of CUBIT activity.

Initially, of course, everything can be very simple, focused upon content and embedded scripts. Gradually things can get more complex but there must be no platform or hardware/software dependence. CUBIT is an abstract machine that can run on many different platforms.

References

[1]
[later]

APPENDIX 1

[for later]

http://mirnova.org

+1 (505) 926-1399 USA +7 (925) 161-6713 (Europe incl. RU)
contact@mirnova.org

Full Contact Information @ website http://mirnova.org

[i] 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.