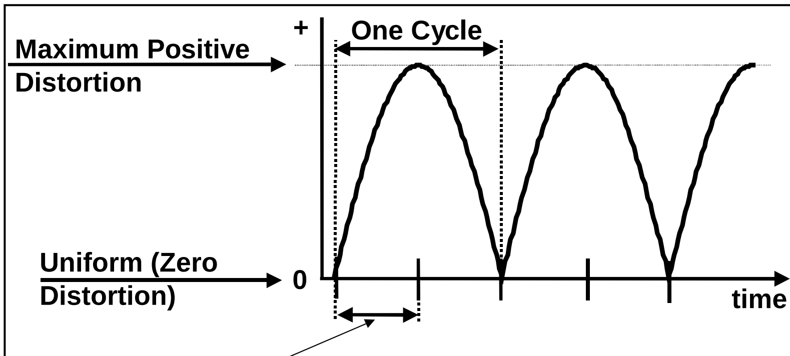
A Complete Set of MBT Figures

Chapter and page numbers taken from the all-in-one-binding paperback

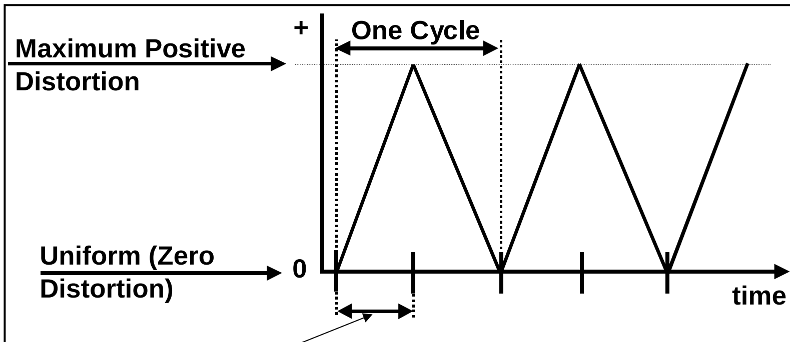
Book 1
Chapter 26
Page 216



A single "time-quantum." The minimum time increment. Also the minimum time required to change the state of one cell from a no distortion (uniform)state to a maximum distortion state. Note that two time-quanta are required to complete one cycle.

Figure 2.1: Unipolar Oscillation in the form of |sin(ω t)|

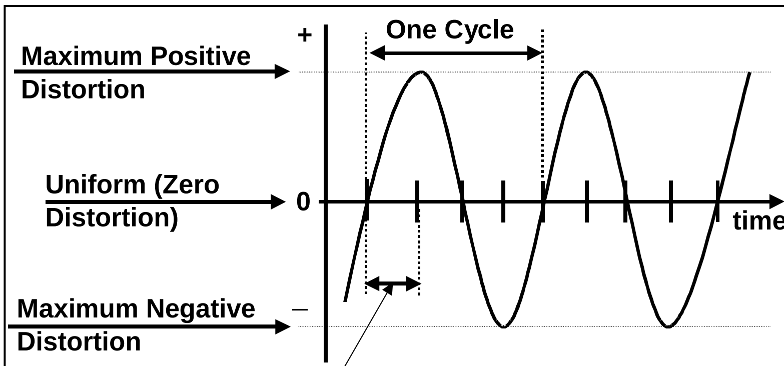
Book 1
Chapter 26
Page 216



A single "time-quantum." The minimum time increment. Also the minimum time required to change the state of one cell from a non-distorted (uniform) state to a maximum distortion state. Note that two time-quanta are required to complete one cycle.

Figure 2.2: Linear Unipolar Oscillation (sawtooth)

Book 1
Chapter 26
Page 217



A single "time-quantum." The minimum time increment. Also the minimum time required to change the state of one cell from a non-distortion (uniform) state to a maximum distortion state. Note that four time-quanta are required to complete one cycle.

Figure 2.3: Bipolar Oscillation in the form of sin(ωt)

Book 3
Chapter 76
Page 654

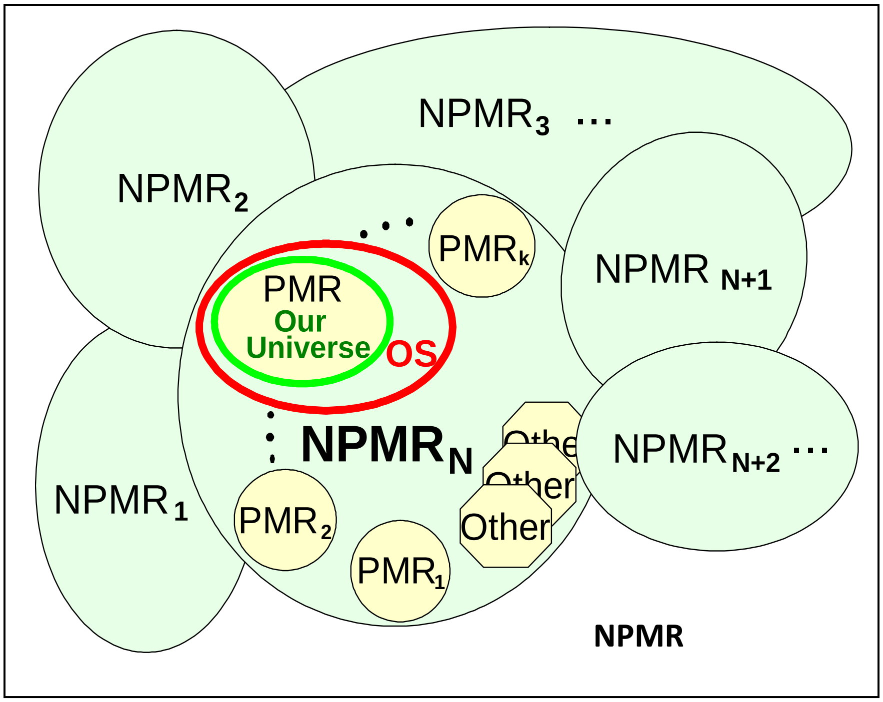


Figure 5-1 Reality Systems: The Big Picture

Book 3
Chapter 79
Page 670

Text Box 5-1: The OS/DELTA-t Loop

1. DELTA-t is incremented.

2. Free will choices, material changes, and energy changes are made, defining a new OS state vector associated with the current DELTA-t.

3. The new OS state vector is compared to the previous one and to the predicted one.

4. All actualized changes between the new and previous OS states are recorded. Predictive algorithms and databases are updated and improved.

5. TBC calculates **M** sequential probable future states of the new OS by running a delta-t sub-loop.

6. Entire OS state vector including all significant possibilities and probabilities is recorded

7. The loop is returned to step 1

Book 3
Chapter 79
Page 673

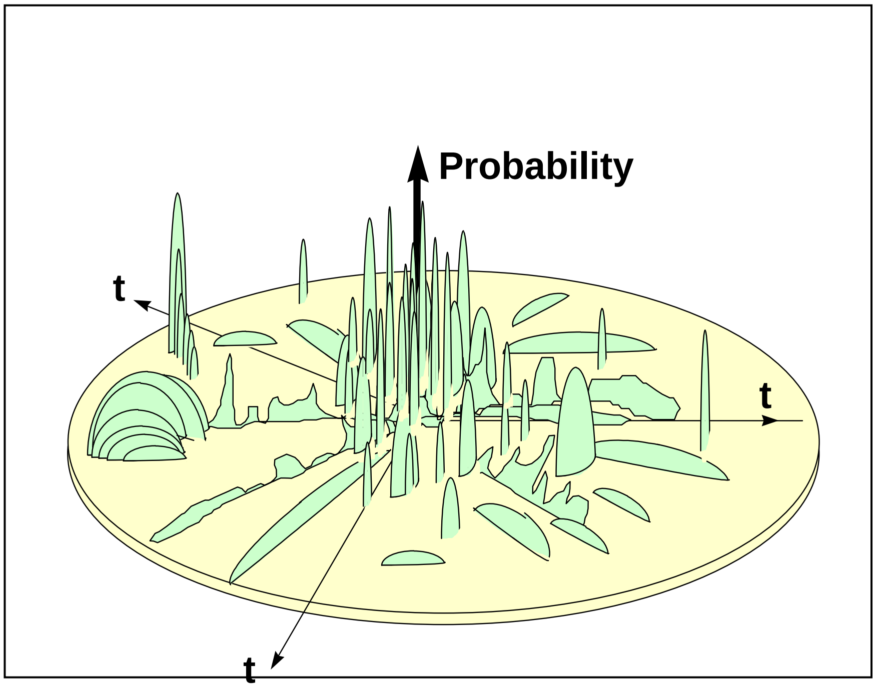


Figure 5-2 Future Probable Reality Surface

Book 3
Chapter 80
Page 683

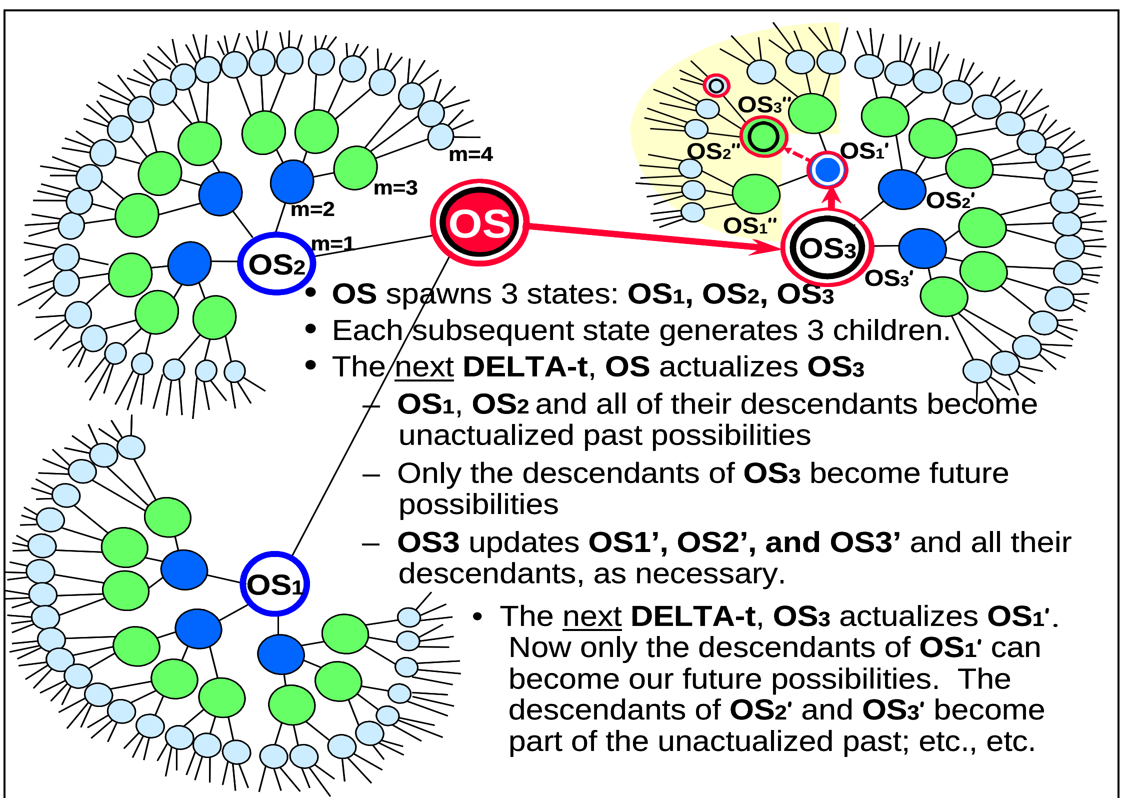


Figure 5-3 Generating the Possible States of OS

Chapters With Equations

Chapters 79 and 80

CHAPTER 79

A Model of Reality and Time

Probable Reality Surfaces
The Simulated Probable Future
Real-time, Our System, State Vectors, and History

1. Probable reality

Let us begin this discussion of time and the mechanics of OS (our collective Big Picture local reality) in the PMR present and slowly work toward a more generalized concept of past and future. Join me here in the present moment and let us see what is happening in $NPMR_N$ to support our sense of reality-present. The Big Computer (TBC) has captured in its database the state of being of OS at this moment. This includes all the objects and energy in the universe as well as all the significant choices that all the relevant sentient beings within OS have at this moment. Later in this chapter we will more specifically, and in more detail, define the set of information that specifies this state of being or "state vector" of Our System (OS).

TBC can now compute everything that could possibly happen next (we will explore this thought more thoroughly later). Additionally, it has accumulated a history file of past behaviors relative to similar choices and can thus compute the likelihood of occurrence (probabilities) of each of the possible things that could happen next. Many of the current choices are dependent on a likely interaction with the choices that all relevant and significant others (including themselves) made the moment (or many moments) before. All possible interactions are defined and evaluated with respect to all possible choices and arrangements of objects and energy, as well as against a complete set of history-based likelihoods (expectation values).

During the time between successive increments of PMRs quantum of time (DELTA-t), TBC has computed OS's probable future – what OS will probably be like during the next **M** (**m** = 1, 2, 3, …**M,** where **M** is an integer) iterations of DELTA-t. Thus within TBC, the dynamic OS has been simulated and its future state, the one that will most likely appear during the next DELTA-t (**m** = 1), has been predicted based on the results of the present state of OS after the last DELTA-t. This OS simulation is run again (**m** = 2) with the results from the previous predictive simulation (**m** = 1) used as input, and the probable outcomes and expectation values for the following DELTA-t are predicted as output.

Each successive output (predicting the state of OS out into the future one more DELTA-t) becomes the input for the next predictive calculation. This process is continued **M** times until TBC has progressed the model of OS out as far in time as it finds useful. The probable OS state vector generated after each iteration (for each value of **m**) during the dynamic simulation of what is most likely to happen in OS during the next DELTA-t is saved in TBC. Remember that we are doing all **M** iterations between actual increments of DELTA-t.

As displayed in Text Box 5-1 below, the iterative process would operate in the following manner. First, a new DELTA-t increment is initiated resulting in the initiation of a new OS state vector. Then, all free will choices and material and energetic changes that define the activity that creates this new OS are made. The choices and changes, once made, define a new and unique state vector (or more simply, "state") of OS that is associated with this particular DELTA-t. TBC now compares the new OS state vector to the previous one and records the actualized changes. It also compares the newly actualized state to the predicted state and makes the necessary adjustments required to improve the accuracy of future predictions.

Text Box 5-1: The OS/DELTA-t Loop

8. DELTA-t is incremented.
9. Free will choices, material changes, and energy changes are made, defining a new OS state vector associated with the current DELTA-t.
10. The new OS state vector is compared to the previous one and to the predicted one.
11. All actualized changes between the new and previous OS states are recorded. Predictive algorithms and databases are updated and improved.
12. TBC calculates **M** sequential probable future states of the new OS by running a delta-t sub-loop.

Next, TBC calculates **M** potential future states (of the new OS state). These **M** calculations (made sequentially one value of **m** at a time) project (simulate) what is most likely to happen during the next **M** increments of DELTA-t. The first (**m** = 1) potential future state (probable reality) of OS is computed in TBC based on the latest actualized (actually happened) input data generated by the choices made during the current state of OS.

This predictive simulation of the state of OS, which progresses by iterating **m** from 1 to **M**, creates a set of sequential probable realities describing the probable future of OS. The subroutine or iterative loop we are using to generate future probable realities of OS must, because it is a dynamic simulation tracking changes, operate on its own time base, and that time base must utilize a much smaller time increment than DELTA-t. This smaller time increment, which we will call delta-t, is associated with a quantum of time in $NPMR_N$. Utilizing a time increment (delta-t) that is very much smaller than the PMR quantum of time (DELTA-t), allows us to model the dynamics of OS (which is a subset of $NPMR_N$) in enough detail so that we can predict the most likely state of OS for each value of **m**. Thus, delta-t is the fundamental quantum of OS simulation time.

The OS state vector simulation runs through its internal calculations using the smaller $NPMR_N$ time quantum (delta-t) until it eventually converges to a predicted future state of OS for every value of **m**. It should be clear that the time increment DELTA-t is composed of or contains some large integer number of $NPMR_N$ time quanta delta-t, and that the computation of the probable future of OS through **M** successive generations occurs between successive increments (DELTA-t) of our PMR real-time.

Now that a set of **M** successive generations of probable future realities has been determined, TBC next records the entire OS state vector representing all significant possibilities and probabilities existing within OS. This step is discussed in further detail in Topic 9 of this chapter. The final step of the OS DELTA-t loop iterates the process by returning the loop back to the first step. DELTA-t is incremented again, and the entire process is repeated for the new DELTA-t.

Let's back our perspective out one more level for a peek at an even bigger picture. Though it lies somewhat beyond our immediate perception, contemplate the concept that delta-t is incremented only after so many ticks of a smaller, more fundamental time increment. We know that delta-t is a small time-increment ($NPMR_N$ time base) used to simulate what is most likely to happen in OS during future DELTA-t time increments. It is used to simulate probable future states of OS and to increment a larger DELTA-t (the OS DELTA-t loop). Furthermore, consider that delta-t is incremented only after so many ticks of a smaller time increment that is used to simulate probable future states of $NPMR_N$ (the $NPMR_N$ delta-t loop).

Because NPMR is an outer loop to $NPMR_N$ (where OS lives), it makes sense that NPMR runs on a smaller time quantum than $NPMR_N$. Thus, just as the OS DELTA-t loop must have the smaller delta-t as its fundamental quantum of simulation time, the $NPMR_N$ delta-t loop must likewise have a smaller time increment than delta-t as its fundamental quantum of simulation time. Keep in mind that NPMR is a superset of $NPMR_N$; also that $NPMR_N$ is a superset of OS and that OS is a superset of our PMR (OS is comprised of our PMR plus a portion of $NPMR_N$).

The fundamental increment of a bigger-picture simulation-time represents an outside loop that provides a larger perspective than $NPMR_N$. The increment of time of such an outside or higher-level loop must be smaller than the fundamental unit of time in $NPMR_N$. Expanding this idea, it is clear that the fundamental increment of NPMR simulation-time may likewise be smaller than the fundamental unit of time in NPMR itself.

Is all this clear or is your head spinning a little? If you are a tad confused, it might be helpful to go back to Chapter 5 of this book and refresh your memory on the subject of incrementing time within simulations. Additionally, a glance at Figure 5-1 (at the beginning of Chapter 4 in this book) and a peek ahead at the discussion of the Even Bigger Computer (EBC) at the end of Chapter 11 might help clarify this bigger picture. Otherwise, if you feel that you mostly get it, absolutely do get it, or don't want to get it any better than you've gotten it, simply go on. In this situation, continuing on (though you find yourself in a light fog) is much better than becoming terminally frustrated. Hang in there, the text gets less technical later.

The predictions produced by calculating the probable state of OS **m**•(DELTA-t) into the probable future become less accurate the further out in time they go. However, because our computer (TBC) and its software are so good, it can progress PMR time out for many years (**M** can be arbitrarily large) in less than a nano-nanosecond.

The result is a PMR space-time event surface in TBC calculation-space. TBC is only a subset of a greater digital mind-space. For we 3D creatures constrained to visualize our mental concepts within an experiential 3D structure, it is easier to think about a planar (two dimensional) event surface extending out in the dimension of simulated time with probability values (of particular events) on the vertical (up) axis – perhaps something similar to the surface shown in Figure 5-2. The horizontal plane, upon which the peaks rest, contain values of time, from t = 0 at the origin to some simulated future time **m**•(DELTA-t), the far edge of the event surface being at the time corresponding to **M**•(DELTA-t).

Events near the present moment typically have the highest probability values (sharp tall peaks). The further out we go in time the flatter the surface gets; peaks tend to broaden and lose height exhibiting very small, rather diffuse, probability values or likelihoods. Nevertheless, there may exist a few well formed and sizable (> .8 expectation value) peaks rather far out in time. You might want to take another peek at Figure 5-2.

In summary, TBC generates a complete set of probable realities covering everything (choices, things, and energy) most likely to happen between each DELTA-t and the next one. TBC then saves and stores these results describing every unique probable state of OS corresponding to each simulated DELTA-t for each value of **m**. This complete set of probable realities going out **M**•(DELTA-t) in our PMR real-time are regenerated after each actual increment of DELTA-t. For the techies among you who are fretting over the apparent inefficiency of such a process, remember that there is no practical constraint on the consumption of computational resources and, as you will find out later, computing and saving **all** potential states generates a database of possibility that supports a plethora of exceptionally useful analysis that needs no additional computation.

I have used the term reality state vector ("reality state," or simply "state" for short) to mean the total description or specification of the state of existing choices, things, and energy that defines OS at the end of a DELTA-t. Later I will define more precisely what a reality state vector is (Topic 9 below) and describe the process that generates it (Chapters 8 through 10 of this book), but first there are a few more basic concepts to be introduced.

2. Introducing real-time – what our clocks measure in PMR

What appears to be real-time is dependent on the relative reality level or loop location of your local perspective within the Big Picture. PMR real-time appears to be continuous but actually progresses incrementally by iterating the small time increment DELTA-t. During DELTA-t, beings, objects, and energy move and change, free will is exercised, and significant choices are made in PMR. Most changes were as predicted by the **m** = 1 calculation of expectation values, but some were not. Adjustments are made. Again, TBC runs the delta-t based OS simulation. Again it re-computes all the probable realities – OS state vector expectation values through **M** generations – creating, updating, and storing the event surface as subsequent calculations progress.

TBC's computing requirements are not as horrendous as one might first think: Clever software finds updating the complete set (**m** = 1 to **m** = **M**) of probable realities to be much easier (merely dealing with changes and errors and their downstream impacts) than re-computing everything from scratch every time. It should be clear that what we sense and measure as time (our real-time) moves forward in PMR by successive increments of DELTA-t, while probable realities (future expectations) are computed within TBC to project the probable future states of OS through **M** successive simulated increments of DELTA-t.

Recall from Section 2 (Chapter 31, Book 1) that the quantum of time in $NPMR_N$ is much smaller than the quantum of time (DELTA-t) in OS and PMR. Thus, during a single DELTA-t of PMR real-time, $NPMR_N$ has many time increments (quanta) ticking away in which calculations can be made, probabilities computed, and probable reality surfaces generated. Also, recall that in Chapter 5 of this book we described how the flow of time in a subset or subroutine of the simulation was dependent on its outer controlling loop and that the simulation could be paused, stopped, and then restarted (relative to the clock in the computer room) without causing any effects within the simulation.

AUM's fundamental clock is the clock on the computer room wall and we are a subset of $NPMR_N$ within a subset of NPMR. In other words, $NPMR_N$ controls PMR's outer controlling loop, while NPMR controls $NPMR_N$'s outer loop. The process just described (the generation of OS, resulting from incrementing our real-time quantum DELTA-t and the calculation of our probable futures) raised to the next level of generality, allows for free will choices among sentient residents of $NPMR_N$. In a similar process to that which generates OS, the free will choices of $NPMR_N$ residents interact with the beings, objects, and energy of $NPMR_N$ to create and actualize successive $NPMR_N$ states of being, which in turn enables the generation of $NPMR_N$'s present and probable future states. As far as I can tell, the digital-state-flipping-AUM-consciousness-bright-awareness-thing directly controls NPMR's outer loop.

3. How the probable reality surface changes

As previously discussed, PMR real-time moves on as DELTA-t is successively iterated. Not only can the expectation values of a future probable state be computed, but the rate of change of these projected probabilities everywhere on that surface can also be computed as a function of DELTA-t. Because these calculations have been made for a large number of DELTA-t, the history of how the probable reality surface has actually changed with respect to real-time is now known. This information (sensitivity of our probability functions to perturbations) can be used to help calculate better, more accurate probable realities. In fact this is exactly what has been going on all along. The probable reality surface represents the most likely future possibilities.

As the present consumes the surface at the origin (t = 0), the surface is extended further into the future at the outer perimeter [t = **M**•(DELTA-t)]. It might be profitable, though simplistic, to imagine the probable reality surface as a circular plane with t = 0 at the center and with t spreading out (increasing value) radially in all directions at once (see Figure 5-2). As real-time marches on, the plane disappears (moves), one DELTA-t at a time, into the pinhole at the origin while a new ring, DELTA-t wide is added to its outer edge to maintain a constant radius of **M**•(DELTA-t). The disk is thus made up of **M** concentric flat rings, with each ring being one DELTA-t wide.

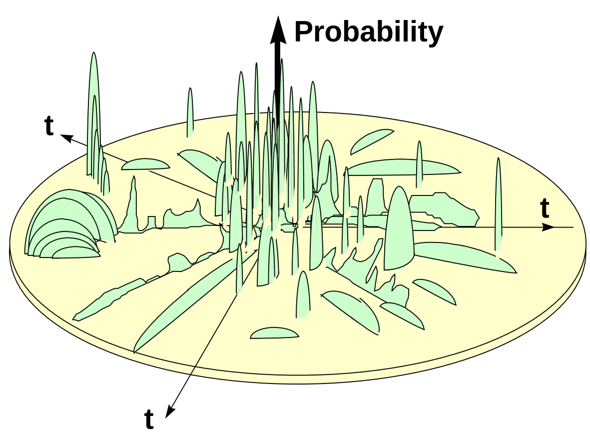


Figure 5-2 Future Probable Reality Surface

Probable Reality Surfaces – The Simulated Probable Future
Real-time, Our System, State Vectors, and History

The pinhole, (more correctly, the mathematical point at t = 0) into which the future probable reality surface is being sucked, represents the present moment. After the present moment, all state vectors are saved. In other words, the present state is defined and subsequently saved to a history file, which contains every previous present state. We will see later how these past (previously actualized) states, captured by saving their present state vectors, remain vital and capable of branching to new potential virtual reality system within TBC's calculation space whenever additional significant input is introduced.

4. Predicting the future

There will be some peaks on the surface that will have relatively large, stable and growing values. Some of these may occasionally occur far out in time (the future). These peaks and the events they represent or relate to would be good bets if you were a prognosticator. The narrower and taller the peak, the more precise the prediction and the more likely the event represented is to occur. Thus, we have future events that can be predicted with good reliability coexisting compatibly with individual free will. From the PMR point of view, intentionally or unintentionally tapping into this database of the most probable possible future events seems to, but does not actually, support the concept of predestination. Free will is required to convert probable events into actual events within the present moment. All information on probable futures is fully accessible from a larger perspective (if your intrinsic noise level is low enough) at: RWW.$NPMR_N$.OS.PMR/probable-reality-database/specific-event/specific-intent.

5. Group futures

Future probable reality surfaces for a particular group, activity, or happening can be computed. Specific summations can be taken over all the choices made by sentient beings and all the changes of objects and energy that have, or are projected to have, an impact or influence on a particular group (family, tribe, organization, corporation, nation, culture, planet, fault line slip, endangered species, rain forest, football team, or human race). The specific group's probable reality surface shows only those probable future events that are significant or related to that group. An individual interested in a specific group's probable future can easily filter all interactions for only those that pertain to that particular group through a process that is analogous to submitting a database query function where your intent designs and executes the query. A view of the collected events that are defined and limited by the properties of your query-intent, along with their associated probability values, are available to you through the RWW net.

Probable Reality Surfaces – The Simulated Probable Future
Real-time, Our System, State Vectors, and History

6. The probable future can change

The rate of change (fluctuations on the surface) of the probability functions for expected events for individuals is much faster than the fluctuations for a large group of individuals. Thus, a nation's future is easier to predict than an individual's (is a more slowly varying and stable surface). The probabilities on the probable-reality surface representing our entire planet change even more slowly, allowing for more accurate prognostication. Think of the probable future of a group or organization as the vector summation of all the probable future components of the individuals that affect that group weighted by their likely significance (impact) on the group.

In general, the larger the system, the more "inertia" it has (the less it can be affected by an individual's free will choices or by small random components within objects and energy), and the more stable and reliably predictable its probable-reality surface becomes. On the RWW net, information about these more clear (larger and more stable signal) future events exhibits a higher signal-to-noise ratio to everyone and therefore the information (likelihood of some particular event) is more accessible to more people. (Know anybody who claims knowledge of future earth changes? By the hundreds!)

7. Constraining the number of required calculations

The computational burden is not as horrendous as it might seem. Most of the possible choices produce degenerate (the same) results and can be quickly dispensed with. Individuals only have influence or impact on a small subset (that may or may not be significant) of the complete set of interactions and choices. Objects, energy, and people with free will are generally more predictable than you might guess – particularly if you have all the historical data.

Only a relative few individuals at any one time (even in any one year or decade) have the potential to influence or produce major effects as a result of their choices (what they do significantly impacts the choices of many others). Additionally, large subsets of beings, energy, and objects may be functionally independent of each other. For example, earth relevant calculations could proceed as an independent set until interaction with specific extraterrestrials (ETs) from elsewhere in our universe occurs. Same for the ETs. Furthermore, there are certain rule-set constraints, such as our PMR laws of physics (things never fall up), which further limit the possibilities.

Despite the mitigating factors of degeneracy, independent sets, and other constraints, computing everything that can happen in the universe and all associated interdependent expectation values (probabilities) is a big job, but it is finite. Fortunately, TBC has no problem performing this task using only a small fraction of its overall capacity.

Probable Reality Surfaces – The Simulated Probable Future
Real-time, Our System, State Vectors, and History

8. Defining Our System (OS) to include all the players

An interesting side issue is that of manipulated choice. The manipulation, leading, predisposing, or nudging of PMR awareness by those aware in $NPMR_N$ is another mechanism through which certain probable outcomes are made more likely than others. In other words, another set of interactions that must be taken into account (as part of the OS calculation space input data), are the actions and free will choices made by those extant in $NPMR_N$, but not in PMR, that directly influence or impact the beings in PMR. This interaction affects the state of OS, and is therefore (by definition) a part of OS.

Some of those large, stable, and growing probability peaks exist because they are being encouraged or manipulated by $NPMR_N$ residents who may have much larger perspectives, much better information, a much clearer sense of the future (a better, bigger picture), and a more accessible knowledge base than PMR residents. Consequently, while some peaks (likely events) simply happen of and by themselves, others are guided. Most are a mixture of both.

TBC calculations relevant to our local reality system (OS) must include all beings, objects, and energy in the $NPMR_N$ superset that have an influence upon, or interaction with, our PMR probable-reality surface (not only those beings, objects and energy that exist within our PMR subset). TBC and its software (which can be clever, and does not have to execute a simple brute force approach) are by design thorough and precise in calculating and tracking the facts, possibilities, and probabilities of Our System (OS). OS creates or actualizes its larger reality through the choices of all its interactive beings (embodied or not) and the randomness of all its interactive objects and energy (physical or nonphysical). Changes must abide by the PMR space-time rule-set, the $NPMR_N$ rule-set, the rules of interaction between PMR and $NPMR_N$, and the psi uncertainty principle. Our history (the history of PMR), from a larger perspective, is a subset of the overall history of OS – as European history is a subset of world history.

9. Reality-system state vectors and our history

During a given real-time increment DELTA-t, beings, objects, and energy may move (or change in some other way) and choices are made to actualize the new present which is contained within the state vector representing that DELTA-t. All potential choices not made remain unactualized potential states (possible realities) and have associated expectation values. The complete state vector of OS containing all actualized and unactualized choices is saved in TBC.

The state vector that defines or represents OS at a given increment of time (DELTA-t) is the total collection of information and data that completely specifies everything that actually did happen and possibly could happen (every significant possibility within that DELTA-t), along with its associated probable-reality surfaces. You will hear more about this later.

Thus, the progression of PMR from one DELTA-t to the next DELTA-t produces or traces a history which is the sum total of all the changes and choices that are actually made or actualized that affect or interact with PMR. This trace becomes an OS history thread representing everything that did happen, or in other words, a sequence of all the states of OS that were actualized during each DELTA-t.

The system is not closed. The system is open; beings, objects and energy can come and go in and out of effective interaction during any DELTA-t. TBC keeps track of, and up with, everything that is significant to (interacts with) OS.

This process and its results define our particular world, our particular history, our particular universe, our particular virtual reality – we who are interacting are all in this together, so to speak. Our choices define, in our view, a collective thread of continuous happening and unfolding generated by beings interacting with each other and with objects and energy.

The OS state vector containing the possibilities and probable reality surfaces not actualized, as well as those that were actualized, is saved at the end of each DELTA-t. For this reason, you can, from an awareness in $NPMR_N$, visit the past, view it, extract information from it (it is on the RWW), and even make changes to it that initiate new calculated arrays of un-actualized past probable realities. We will discuss this in more detail later.

10. History, still vital after all these years

You can interact with the actualized as well as the potential non-actualized past. When you interact with any part of it in such a way as to modify it (introduce a new being, new things, new energy, or a new configuration of old things, or change a significant choice or action), a new set of probable futures is computed that incorporates the changes as new initial conditions. A new set of probable reality calculations can now be progressed, creating a new branch, within the non-actualized past database. The nature of this process is like making a copy of a file or simulation program so that you can play what-ifs with it without disturbing the original.

Any point along the OS timeline, actualized or not, is a potential branch point, but branches do not spontaneously sprout from every point – they occur only when a significant change in the reality state vector is produced by defining a new and unique set of potentialities. If the change creating new initial conditions is trivial as evidenced by no significant change in the future probable reality surfaces for all reasonable possibilities (not only the most likely ones), then that branch degenerates back to the initial point of departure. Adding a new electron to the system or changing an irrelevant choice, therefore, does not start a new parallel reality in the what-if calculation space of TBC. More of our choices than you would probably guess are irrelevant in the interactive Big Picture.

In summary, I have described what happens when a change is made to any part or detail of the complete set of everything that could possibly happen, which is computed at the end of each real-time (PMR-time) DELTA-t based on the history of all the beings, objects and energy and on all possible configurations or choices. Because of the small size of DELTA-t, our PMR history appears to be a continuous thread traced by the collective result of the interactions and choices taken, experienced, or actualized. What has not been actualized up to this point has simply been saved. However, both non-actualized past and probable future states remain operationally viable allowing the state vector database to be queried and what-if simulations to be executed by intent.

CHAPTER 80

A Model of Reality and Time

Using the Delta-t Loop to Project (Simulate) Everything Significant That Could Possibly Happen In OS

Let's generalize and broaden our model by looking at the possibility that everything significant that can happen does happen. This is a key concept to understanding the breadth of our multi-dimensional reality, and to appreciating how AUM optimizes the output of its consciousness experiments by collecting data and amassing statistics that describe all possibilities simultaneously.

In the previous chapter, I described the complete set of state vectors representing everything that will most likely happen in OS. This was computed by incrementing delta-t (simulation time) through **M** consecutive iterations in between each increment of DELTA-t (PMR time). Recall that as **m** progresses from **m** = 1 to **m** = **M**, the delta-t loop converges upon the most probable future state. This was accomplished by evaluating all the possible future states in order to determine the most probable one. The most probable future state for that iteration of **m** then becomes part of the set of stored OS probable reality state vectors. Now TBC is going to track and store every significant possible future state (and its associated expectation value) that is evaluated for each iteration of **m**, not only the most probable one.

In order to assess all the significant possibilities, our understanding of the delta-t loop must be expanded. A more generalized delta-t loop process must now not only compute the most likely future states, but also track all (regardless of their likelihood) possible significant future states for each iteration **m** = 1, 2, 3,… **M**. Furthermore, each of these possible future states is assigned an expectation value that is a measure of its likelihood of being actualized. The mechanics and implications of this broadened delta-t loop functionality are discussed in detail in the remainder of this chapter, and are illustrated in Figure 5-3.

Before continuing with the description of this new application of the delta-t loop, I want to define the concept of significant states. A significant state is one that represents some unique, viable, meaningful configuration of OS, even if it is perhaps somewhat unlikely. Essentially, TBC generates significant states by computing all the permutations and combinations of all the free will choices, all the potential changes in objects, all the energy state changes, and then eliminates the redundant or insignificant states. All significant states with a probability of actualization above some small arbitrary value are enumerated.

For a given value of the iteration index **m**, the total number of significant possible future states is not known until after they have been generated. Thus, as the delta-t loop is iterated, there may be a different total number of significant possible (though not necessarily probable) future states for each specific iteration **m**. Recall however that for each specific iteration of **m**, only one of the significant possible future states will eventually be actualized and take its place on our seemingly continuous OS-PMR history thread. The state that eventually becomes actualized will probably be the one that was previously given the highest probability of being actualized – but not necessarily. The collective free will is free to choose whatever it will; updates and adjustments are made as needed to accommodate the vagaries of free will.

Perhaps the simplest way to think of this generalized process is to imagine that a dimension of width has been added to the information recorded during each iteration **m** of the delta-t loop. Look ahead to Figure 5-3: The example given shows parent-child state generation exhibiting geometric growth. During the first iteration of **m** (**m** = 1) there are three significant possible futures states generated (OS_1, OS_2, OS_3), including the one determined to be most probable (double bordered OS_3). TBC tracks and stores all three states associated with iteration **m** = 1. Here we have chosen the small number three (**M** = 3) to make our visualization easier to grasp and present graphically. In actuality, there is a very large number of significant possible future states. Every circle in Figure 5-3 represents a unique state vector of OS.

This newly expanded function of the delta-t loop (tracking all possible states instead of just the most likely ones) represents a generalized larger view of our previous understanding. As such, it is more complicated to describe and to follow. Referring to the simple example given in Figure 5-3 may help provide a better understanding. The next step (**m** = 2) of this generalized $NPMR_{\mathbf{N}}$ delta-t loop is to project (simulate) a set of possible (though not necessarily probable) future states for each of the alternative states generated for **m** = 1. That is to say that each significant possible future state that is generated during iteration **m** spawns another complete set of significant possible future states during the next value of **m** (and so on as **m** is sequentially stepped to **M**). For example, in Figure 5-3 each nearly solid black state (**m** = 2) generates three medium gray (**m** = 3) states, which each generate three light gray states (**m** = 4).

Using the Delta-t Loop to Project (Simulate) Everything Significant That Could Possibly Happen In OS

The result is a geometrically expanding array of significant possible future states originating with the current OS and iterating **M** generations into the simulated (projected) future – representing a total elapsed time of **M**•(DELTA-t).

In summary, during iteration **m** = 2, each of the previously generated (first generation) alternative states will generate some number of significant possible (second generation) future states of its own. The generalized delta-t loop is then recursively applied to each second-generation alternative state. This process continues until the delta-t loop has projected all significant possible future states of OS for **m** = 1, 2, 3, …**M** iterations by simulating (projecting) everything significant that could possibly happen (above a certain level of expectation) during **M** consecutive increments of DELTA-t.

We are no longer working exclusively with what is most likely to become actualized and what has previously been actualized (our PMR-OS history). We have now formed a super-set that includes all that, as well as some significant (worth following) states that will not and did not happen. In other words, a larger, broader set of states defining all significant possibilities of OS has been formed. Mechanically, this was accomplished by expanding the scope of the delta-t loop to recursively enumerate and determine all the significant possibilities of OS.

This enumeration and determination of potentially significant state vectors does not need to be accomplished by computational brute force. Given that the Fundamental Process would unquestionably need to create extremely clever evaluative operating systems and software for TBC as it functionally evolved within AUM's consciousness, we can assume a certain efficiency of process is achieved. After all, evolution is the unparalleled master of developing efficient and effective processes within each specific operating environment. For example, such

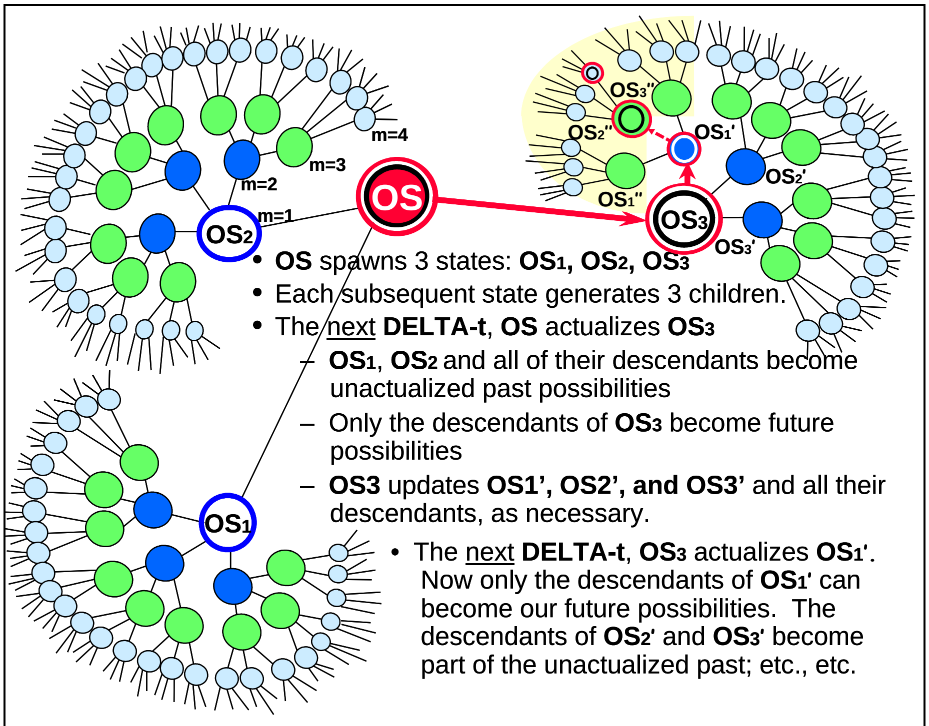


Figure 5-3 Generating the Possible States of OS

software could be used to remove all the extremely unlikely, insignificant, uninteresting, unproductive, degenerate, duplicative, repetitive, meaningless, and useless states to form a complete set of useful alternate reality state vectors specifying <u>everything significant</u> that possibly could happen in OS during the next actual DELTA-t. Recall Topic 4 in Chapter 4 of this book for a short list of overall goals that suggests some of the evaluation criteria this expert system software might use to make decisions. Remember, the evaluative processes do not have to be perfect – the final results need only be functionally adequate and statistically meaningful – perfect calculations and processes are never required.

TBC calculates the probability that each projected possible OS state vector might be actualized by the free will choices and changes in objects and energy that will be made during the next actual increment of DELTA-t. The one that is most likely to be actualized becomes the first point (**m** = 1) on the future probable reality surface of OS that we discussed in the previous chapter. In the example shown in Figure 5-3, the states that will eventually be actualized are double bordered. OS$_3$ most likely, but not necessarily, represents the first (**m** = 1) flat ring on OS's <u>most probable</u> future reality surface. (Reference Figure 5-2 located in Chapter 7 at the end of Topic 3 in this book.) OS$_2$ and OS$_1$ are also **m** = 1 states but remain unactualized – the choices they represent were not chosen by the collected free will actions of the sentient beings in OS during that actual DELTA-t.

The next pass (**m** = 2: solid black circles) through this generalized OS DELTA-t simulator will allow <u>each</u> of the possible **m** = 1 states (both likely and unlikely) to likewise generate all the possible significant states that could be generated from the initial conditions that this particular state vector represents. Again, the probabilities of actualization are computed for <u>every</u> state vector generated. For example, only one state generated by OS$_3$ can be most probable and take its place on the most probable future reality surface of OS$_3$. Also only one state (depicted by double bordered OS$_1$' in Figure 5-3) generated by OS$_3$ will be actualized by the free will of the sentient beings within OS$_3$. This process, repeated **M** times (once for each value of **m**) gets to be a mind full. I am afraid that we will need to resort to a generalized subscript notation to keep this mental picture focused.

Let's take it from the top utilizing subscripts to form a generalized description of this process. Each alternate reality state vector, differentiated by the index **i**, represents the state vector defining OS$_i$. The subscript **i** keeps track of as many unique state vectors (**i** = 1, 2, 3, …) as there are unique arrangements of sentient choices and other variables (objects, position, or energy). These OS$_i$ state vectors constitute parallel, possible, or potential future realities of OS or from a PMR-centric view, parallel, possible, or <u>potential</u> future universes. Sometimes travelers in NPMR$_{\mathbf{N}}$ get into these parallel realities (such as the never-to-be-actualized solid dark (m = 2) or medium gray (m =3) circles attached to OS$_1$ or OS$_2$ – or the could-be-actualized light gray (m = 4) outermost circles attached to OS$_3$''), and fail to realize they represent reality states that were or are merely possible and not necessarily probable. A measure of each state's probability of being actualized is available, but needs to be accessed with a separate intent. In other words, the probability of actualization does not automatically come integrated with the experience of the reality – you have to ask a separate question and be precise with your initial intent.

Each of the alternate reality state vectors described above becomes the starting point for another. Everything significant that could possibly happen is computed based on the unique permutations and combinations of all the possible states of objects, energy, and the free will choices of beings originating from the particular initial conditions of each particular alternate reality state vector.

Thus, parent OS_i (first generation) state vector possibilities spawn new child OS_i' state vectors (second generation), which in turn spawns yet a new generation of child OS_i'' state vectors (third generation). This progression continues so on and so forth, until one has progressed this family tree of possible states (all offspring of the original OS and all computed during the single time increment DELTA-t) through **M** generations. Remember, **M** is an arbitrarily large but finite integer.

Every alternate state vector (every circle in Figure 5-3) can generate its own probable reality surface of expectation values by tracing the states that are most probable from generation to generation of its descendants. (Note: **M** is finite because this is a real process generating the real reality that we presently live in. This does not represent an imaginary or theoretical process – it is a practical model of how the larger reality operates.)

In this way, the number of unique and useful alternate reality system state vectors grows (removing all useless and redundant states) until we arrive at a complete set of alternate reality system state vectors representing everything that could uniquely and usefully (significantly) happen. All the state vectors that result from this progression have been derived (originated) from the OS state vector during this present DELTA-t (our present moment).

During the next DELTA-t, one (and only one) of these possible states (first generation OS_i) becomes actualized (through our free will choices and the changing objects and energy) as our next present moment. The descendants of that state (the OS_i that was actualized) become unactualized future possibilities, while the descendants of all the other non-actualized OS_i (a much larger group) become unactualized past possibilities. TBC saves and stores everything.

The unactualized future possibilities have a finite chance of being actualized at some time in the future, while the unactualized past possibilities can now never be actualized by free will choices within OS. Now integrate this picture with the one described in the previous chapter. If, from the set of simulated unactualized future possibilities you trace the single most likely state to be actualized within each of the **M** generations, you would have defined the probable reality surface for OS that was defined in Chapter 7 of this book.

The word "actualized," as it is used here, refers to what actually happened or actually took place from our perspective – the perspective of OS. States are actualized and reality is created in the present moment. For us, it is created DELTA-t by DELTA-t – one increment (fundamental time quantum of PMR) at a time. The history of OS is the sequential record of the actualized present moments of OS.

Because time is quantized, and TBC has a good memory, both the actualized and unactualized states (state vectors) can be saved. Every saved state vector is as complete, vital, and capable of generating new states as any other saved state vector within TBC. The set of state vectors within the group called unactualized past possibilities are not dead states; they are simply dormant states. They are as alive and vital as any – they have simply not been actualized or chosen by the free will choices and changes of objects and energy that define the dynamic history thread of Our System of reality.

A short aside is in order here. I can hear you wondering:

"Granted, every state vector is theoretically capable of generating new states, but why would an unactualized state do this?"

You are absolutely correct: It wouldn't change spontaneously. It would simply sit there with all its possibilities laid out for **M** generations, unless something changed in its defining choice-set. For example, a sentient being could travel back through history to that particular state vector and alter something significant, thus modifying that state vector and all its descendants. If the change represented one of the possible choices previously considered (highly likely), no new calculations are necessary, otherwise a new branch is generated and a new larger set of possibilities would be created.

However, even if a new array were generated, once all its possibilities were filled out it would simply sit there until additional unique significant changes were introduced. There is no need to continue making calculations on unactualized states of a particular OS. Unactualized states do not require much updating – unless somebody with free will is introducing new significant initial conditions and creating new branches within the old set of possibilities. They simply sit there as a mostly static complete array (database) of the possibilities and retain the <u>potential</u> to branch (calculate new possibilities) if new initial conditions are introduced.

Using the Delta-t Loop to Project (Simulate) Everything Significant That Could Possibly Happen In OS

> This arrangement (allowing for unique input while maintaining an exhaustive database of possibilities) enables the running of what-if analysis to ascertain the impact of having made specific choices. This type of analysis is often used as an aid to help certain sentient beings that are between physical manifestations in PMR to understand the implications of previous choices and overcome personal belief systems. Such analysis is a typical part of the planning process for more aware beings trying to learn as much as possible from their past experiences before initiating another PMR experience.
>
> For those who do not understand the larger reality well enough or do not have the necessary control, others typically guide this process for them. This analysis capability is generally available to any $NPMR_N$ or PMR being who is sufficiently aware and in control of their mental faculties and intent within $NPMR_N$. *

Now that DELTA-t has incremented and an OS_i state has been actualized as the current OS state, the second generation $OS_{i'}$ that are children of the actualized OS_i state, become first generation possibilities to the present actualized moment. Again, **M** generations beyond the present state are computed for all states (actualized and unactualized) that contain significant possibilities. And so on, and so forth, this process marches on, generating and computing potential reality-system state vectors describing everything significant that might happen, along with everything significant that might have happened (but didn't, from our point of view). TBC saves every state vector, its genealogy, and its likelihood or probability (relative to its siblings) of being actualized by its parent.

The past of OS is represented by a particular solid thread connecting all our past actualized states as it meanders through the matrix of all past possibilities. Our perceived past or history can also be described as a specific sequenced subset (previously actualized states) of the all-past-states database. Likewise, our probable future is represented by a dashed thread snaking through a vastly smaller database of future possible states, picking out only the states with the highest likelihood of actualization, as it moves sequentially from generation to generation. This future thread represents the most probable reality or probable future surface of OS (see Topics 1 and 8 in Chapter 7 of this book).

We have described and generated a set of state vectors representing everything significant that can happen (including everything significant that might have happened and everything significant that might happen yet) through **M** generations beyond the original common ancestor OS. All this is calculated between each DELTA-t. Because this concept is complex, let us summarize quickly before continuing.

Using the Delta-t Loop to Project (Simulate) Everything Significant That Could Possibly Happen In OS

Previously, we developed probable reality surfaces for **M** sequential simulated increments of DELTA-t. These probable reality surfaces only described everything most likely to happen. Now, we have broadened that concept by describing everything significant that could happen. The process starts with OS at a particular DELTA-t and projects (simulates) **M** generations of possible future significant states of OS. This is accomplished between successive actual DELTA-t increments (real-time is standing still in OS). It is accomplished by incrementing the $NPMR_N$ delta-t loop, which, at each iteration **m**, generates all the significant possibilities (the OS_i) for each of the OS alternative states previously defined. This delta-t process continues through **M** iterations, progressing and expanding to project or simulate everything significant that could possibly happen during the next **M** iterations of DELTA-t.

During the next actual DELTA-t (real-time in OS moves forward one increment), one and only one of the OS_i states, is actualized to become our present moment. The actualization of only one state leaves a large set of unactualized past possibilities. Every state not connected upstream to our newly actualized present state becomes an unactualized past possibility. In other words, only those relatively few states that are descended from the just actualized (our new present) state now make up our future possibilities. TBC always maintains a calculation space of **M** generations beyond the present. (We will generalize the concept of **M** later but it will serve us well in the meantime to think of it as a fixed integer.) As this process continues, redundant states among the unactualized past possibilities are collapsed. Entire branches of this family tree may cease to expand for lack of further significant unique possibilities.

The initial massive calculation (running the delta-t loop to generate every significant possible future state through **M** generations) must be done only once (say during the first increment – the beginning of time for OS). Other than a few relatively minor adjustments that may need to be applied to the previously generated states (allowing for unforeseen changes in initial conditions and imperfections in TBC's evaluative and predictive software), all that remains is the creation of the newest generation of children states.

Every actualized reality system state vector, flourishing and evolving within its own dimension, represents a dynamic open (entities and objects can come and go) reality with an active copy of you and everyone else (including all the objects and energy) it inherited from its parent (along with all the pending potential choices, interactions, and conditions).

It may be helpful here to point out that dimension is to TBC as a line on a sheet of paper is to us. Or better yet, as the text-line on a computer screen is to us – simply press the enter key to get a new one. Those analogies are not perfect. Perhaps a better one would be that dimension is to TBC as a saved file in our computer is to us. You get the idea. TBC spins off a new computationally alive dimension within Our System's multi-dimensional reality for every uniquely significant reality system state vector it generates.

In a bigger picture, each dimensioned local reality (the various PMR_k for example) describes a diverging, branching, set of uniquely dimensioned potential worlds. Think of saved files that may contain sub-files – folders within folders – with each folder or sub-file containing an executing piece of the overall simulation. Each reality exists within a unique dimension, folder, or memory space within TBC. All these realities existing within their various dimensions or dynamic folders are computationally alive (can be modified) subsets of a larger simulation, expanding into their potential futures by the beat of their <u>own</u> time artificially constructed or simulated by successive increments of DELTA-t_k. OS is one of those local realities – the one that we sentient beings in OS have collectively chosen to actualize. I will discuss this subject again from a slightly different viewpoint in Topic 3, Chapter 11 of this book where we will again contemplate an Even Bigger Computer (EBC) and multiple PMR_k.

We have thus constructed a process to support everything significant that can happen, in fact, does happen – at least in TBC calculation space. Nevertheless, everything that can happen is not actualized. If, in the rare instance (allowing for imperfection in TBC's software) where the state that is actualized is not (to a significant degree) one of the previously generated OS_i, then it is simply added to the set of OS_i.

Although TBC's software can be exceptionally clever and efficient, it does not have to be perfect. Perfect processes, like infinite processes, are unnecessary to the development of this model. We are talking about real processes here, processes that are imperfect and finite. There is a finite number of beings, objects, and energy states among objects, and each of these has a finite number of choices and ways to change. All the significant permutations and combinations of all the possibilities through all **M** generations is probably an extremely large number (especially from our PMR perspective), but it is finite and consumes only a tiny fraction of the capacity of an apparently infinite (but actually finite) AUM.