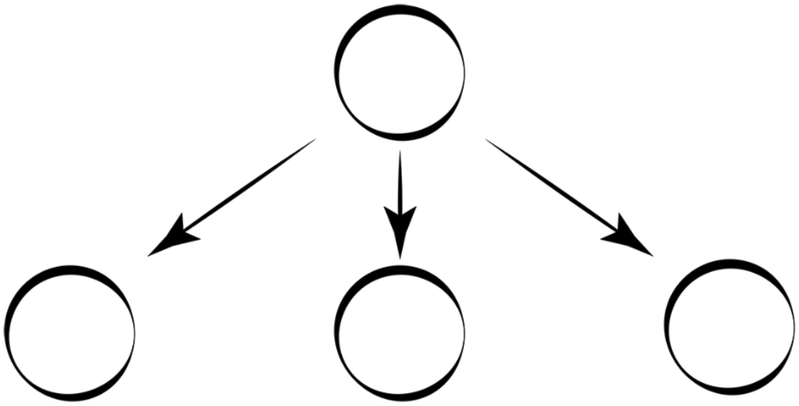
THE SECRET LIFE OF NODES

by Justin Alexander – September 26th, 2020



Node-Based Scenario Design describes an ultra-flexible structure for designing scenarios based primarily around the acquisition and flow of knowledge: When the PCs learn about a node, they can go to that node and, usually, find information that will let them learn about more nodes (which they can then go to and find more information).

The structure is extensible, very robust, and lends itself naturally to mystery scenarios, which are based around the acquisition of knowledge – i.e., the information you gain in a node comes in the form of clues (or leads) that point to other nodes where you can continue the investigation. However, GMs who use node-based scenario design often find its usefulness extends far beyond simply traditional mystery stories, as it turns out that the basic concept of "things which are connected to each other" has a fairly broad application.

The original essay included some simple examples of how node-based scenario design works (most notably the Las Vegas CTU scenario in Part 3 of the series), but those really only scratched the surface of what you can do with the structure. Plus, I've spent another ten years learning new tricks and techniques. So now it's time for a little applied theory as we dive into some examples of how the node-based structure can be used in actual practice.

REALLY PRACTICAL EXAMPLES

If you just want to see full scenarios featuring node-based scenario design, here's a quick reading list.

The Masks of Nyalathotep is a classic *Call of Cthulhu* campaign by Larry DiTillio & Lynn Willis. It had a major influence on codifying the Three Clue Rule and is also a great ante-example of node-based scenario design at a campaign-wide level.

Eternal Lies by Will Hindmarch, Jeff Tidball, and Jeremy Keller is a *Trail of Cthulhu* campaign inspired by *The Masks of Nyalarthotep*. I did an Alexandrian Remix of the campaign, which is probably one of the most comprehensive examples of how I prep a node-based campaign currently available.

Quantronic Heat is a three-part adventure series for the *Infinity Roleplaying Game* designed by Nick Bate and myself (mostly Nick). The first adventure, "Conception," in particular, is a very clean example of scenario-level node-based design.

Here on the Alexandrian, you can check out *The Strange: Violet Spiral Gambit*. This is another very clean example of scenario-level node-based design, serving as a pretty good Platonic ideal of what a node-based mystery scenario looks like. Other examples on the site include *Gamma World: The Egyptian Incursion*, *Star Wars: Red Peace*, and *Eclipse Phase: Psi-chosis*.

Welcome to the Island is an adventure anthology for *Over the Edge* that I was lead developer for. The scenario I co-wrote for the book ("Seversen's Mysterious Estate," with Jonathan Tweet) isn't node-based (it uses the party-planning scenario structure), but most of the other scenarios in the book do use node-based design, often adulterated with other scenario structures. For example, Jeremy Tuohy's "Battle of the Bands" has the PCs trying to reunite a pop-punk band from the mid-2000s, with each band member being a node that they need to track down.

NODES VS. REVELATION LISTS

What is a node?

In Part 9 of *Node-Based Scenario Design* I list five types of nodes:

- Locations
- Characters
- Organizations
- Events
- Activities

Activities have proven to be exceedingly rare in my scenario design (maybe your mileage will vary). In any case, I tend to think of **nodes** as things that the PCs can go and interact with.

PROACTIVE NODES

There are also **proactive nodes** which typically do the opposite: They come *to* the PCs and interact with them.

Proactive nodes are usually characters. I often think of them as the "response teams" of a scenario. This is frequently literal, with my proactive nodes being the people who are sent to deal with the PCs once the bad guys realize they're snooping around. But it's also a useful metaphor: Proactive nodes are the easiest tools I can use as the GM to take action at the table instead of simply reacting.

What about other proactive nodes?

Well, locations tend to be stationary and rarely follow people around… unless you apply sufficiently advanced technology and/or magic. The technology doesn't really need to be all *that* advanced, either: An abandoned ghost ship drifting into the town's harbor can easily be a proactive node.

Organizations can be proactive in their responses, but the response usually takes the form of a specific response team (which would usually be the actual, character-based proactive node).

Events can be proactive if they're of a sufficiently large scale *or* if they're specifically targeted at the PCs: A city-wide festival or a ritual that causes all the trees in the forest to start weeping blood count as the former. A series of enigmatic phone calls that ring up a PC's cellphone would be an example of the latter.

Almost anything can be a proactive node if it can be framed as information being pushed on the PCs. For example, if the PCs are cops investigating a series of bank robberies, they would be automatically notified if a new bank robbery took place. (Is this an event or a location? Both? Does it matter? Don't get too obsessed with the categories here.)

In addition to creating a living world where there are both long-term and short-term reactions to the PCs' choices, proactive nodes also have the practical function of getting a scenario back on track. For example, if the PCs have missed all the clues pointing to a particular revelation, a proactive node can be used to provide a fresh lead.

REVELATION LISTS

On that note, a **revelation** is any conclusion that the PCs can make or need to make in the scenario (see the *Three Clue Rule* and *Using Revelation Lists* for more information).

A revelation can be a node (i.e., "we need to go check out that location / character / organization / event"), but it can also be stuff that isn't a node (i.e., "the runes are Achaean" or "the rage-wraiths can be kept at bay by the smoke of burning wisteria").

In my more recent work, I've started talking about clues & leads, where leads point you to places where you can continue investigating (i.e., new nodes) and clues point to other types of revelations (often the solution to the mystery; e.g., Bob's the killer). In practice, though, these can often overlap. "Bob's the killer" is a mystery solution which also points at Bob (who, as a character, might be thought of as a node). Furthermore, for a satisfying mystery you'd probably want to establish Bob before revealing he's a killer, so there's a "Bob exists / is here / is involved" revelation that points to Bob-as-a-node and then a "Bob did it" revelation that points to the Bob-as-solution.

I think it's useful to separate the node-based list of revelations from the list of other revelations. I just find this to be conceptually clearer. (This assumes that there ARE non-node revelations. That won't always be the case. Pure procedurals and breadcrumb-style mysteries often have a 100% overlap between revelations and nodes. This also tends to be true for non-mystery scenarios.)

In any case, your node list will most often be simply copy-pasted into your revelation list, with a couple notable exceptions:

The **initial node** that begins a scenario almost never requires a revelation list. The exception are scenarios with multiple points of entry, in which case you may want the investigation from one **entry node** to potentially lead to other entry nodes. (This is also only true within the context of the scenario itself: The PCs might actually be led to the initial node of a scenario by the revelation list of another scenario. More on that when we discuss fractal nodes and node-based campaigns below.)

Proactive nodes usually don't have a revelation list, either. These nodes are designed to come looking for the PCs, after all, so they don't necessarily require a mechanism that will lead the PCs to them. (Many nodes, of course, could be *both*. I tend to just list those as normal nodes and then opportunistically play them as proactive when or if it comes up.) What I will do for proactive nodes, though, is include a list of them in my prep notes with the likely **triggers** that will cause them to take action and/or become available for action. These triggers can include stuff like:

- Monitoring Avery Hall [and might therefore spot the PCs investigating Avery]
- Sent to kill Avery Hall [potentially while the PCs are talking to Avery]
- Staking out Node 1, Node 2, or Node 6
- Notified by Avery [if she becomes aware of the PCs]
- Sent by Avery if she learns they know about the jade statue
- Trigger randomly as the PCs are traveling between locations
- Respond to any violence in the neighborhood in 1d6+10 minutes

These triggers aren't really a binding list. They're more of a quick personal reminder for how particular tools can be used.

Node-Based Campaigns

ETERNAL LIES
HINDMARCH, TIDBALL, AND KELLER
PELGT34
Pelgrane Press
The Masks of Nyarlathotep Companion
Chaosium

When talking about node-based design, I've generally used examples of single scenarios (and usually quite simple scenarios). This has the advantage of keeping the examples relatively uncluttered, but can also be inadvertently deceptive by masking the true power of node-based design.

For example, nodes can be location, characters, organizations, and so forth. But conceptually they can also be *entire scenarios*.

This is usually how I design campaigns: I start by brainstorming the major scenarios that are part of the campaign. I think of each scenario as a node and I connect these nodes with clues, following the standard Three Clue Rule, Inverted Three Clue Rule, and all of that. For the sake of argument, let's call these **campaign nodes**.

Once I've sketched out that broad plan, I can look at each individual scenario and work it up in detail. Many of these scenarios will also be node-based (featuring **scenario nodes**), although it's easy enough to mix in other scenario structures (like dungeons or raids or parties or whatever) as appropriate.

The *5-Node Mystery* essay has an example of this process. The 5-Node Mystery itself is a basic node template for creating a simple mystery scenario with five nodes – an initial node, three locations/people, and a finale arranged into a simple pattern. You can whip up five of these simple mystery scenarios and then arrange them *in the exact same pattern* as campaign nodes to form a full campaign.

You can also invert this process, starting with a single scenario, linking it to other scenarios, then linking those scenarios to still more scenarios, organically building up a node-based campaign. (This can obviously work well as a way of building an active campaign week-to-week as you play.)

> *Note: Not every campaign node necessarily needs to be a full scenario. You might have a bunch of campaign nodes that are full scenarios (unfolding into much greater specificity) connected to another node that's simply "Admiral Dawson" (a single NPC).*

The Campaign Binder

My campaign notes generally live in a binder. The first section of this binder contains all the campaign-wide details — lists of campaign nodes, revelation lists, etc. Then each additional section is an individual scenario or similar chunk of information.

To keep the individual scenarios / campaign nodes organized, I'll frequently number them. They're usually also sorted into conceptual bundles. You can see an example of this in Part 7 of the *Dragon Heist Remix*:

0.0 Campaign Overview
1.0 Finding Floon
2.0 Trollskull
3.0 Nimblewright Investigation
3.1 Gralhund Villa
4.1 Faction Response Teams
4.2 Faction Outposts
5.0 Heist Overview
5.1 Bregan D'Aerthe – Sea Maidens Faire
5.2 Cassalanter Estate
5.3 Xanathar's Lair
5.4 Zhentarim – Kolat Towers
6.0 Brandath Crypts
6.1 The Vault

To briefly summarize:

- The 0.0 document is the campaign-wide material.
- Node-based design mostly kicks in with Part 3.0, which is a node-based mystery in which the solution is going to Part 3.1 (which is a raid scenario).
- The 3.1 raid contains a bunch of leads pointing to other scenarios. I've broadly organized these into Part 4 (faction-related stuff built as a sort of node campaign inside the node campaign), Part 5 (the major heists, which the PCs are pointed to by leads in Part 4), and Part 6 (the final scenario, which is built as a dungeon crawl and further broken into two parts for convenience).

Sometimes this conceptual structure is literally just the nodes themselves in a simple list. *Masks of Nyarlathotep* and *Eternal Lies* use a globetrotting structure to similar effect: Each major city visited by the players is treated as a separate scenario (i.e., a campaign node) with links between them.

Act II of my *Ptolus: In the Shadow of the Spire* campaign has forty-two different scenarios. In addition to conceptually breaking the campaign into acts, I have further divided this act of the campaign into three sections, using prefixes to distinguish them. (So this campaign features adventures with organizing codes like NOD4, CC3, and BW09. Which may sound confusing, but is perfectly clear in context.)

My big point here is that there's not necessarily a "one true way" here. It's what works for you and it's what works for a specific campaign.

Using Nodes to Manage Prep

The conceptual separation of a campaign into separate nodes is also a great way to manage prep and, in accord with the principles of smart prep, minimize wasted prep.

Basically, you can set up the essential elements of a campaign node (in terms of how it connects and interacts with other campaign nodes) without fully prepping the scenario that lies "behind" that campaign node. Therefore, you don't actually need to prep that scenario until the PCs move to engage it.

For example, when I originally laid out Act I of *Ptolus: In the Shadow of the Spire*, it featured thirteen scenarios. Because of the choices my players made, however, I ended up only needing to prep seven of those scenarios.

The high-level view of the campaign gives enough detail to have a complicated web of interrelated content, while also conceptually firewalling the campaign in a way that prevents you from pouring a bunch of extra detail into areas where it turns out you don't need it.

> *Note: As you design and expand individual campaign nodes, you'll often find new opportunities to link the node to other nodes arising naturally out of the material. Do so. As a corollary of the Three Clue Rule suggests: More clues are always better. (Make sure to list these new connections in your campaign-wide documentation, too.)*

ADDING NODES

The modular nature of node-based design, however, also makes it easy to integrate new nodes into your campaign structure.

In fact, it's not unusual in the slightest for the actions of the PCs to generate new nodes by either creating leads out of whole cloth (by pursuing methods of investigation or a strategy of action that I hadn't anticipated) or following leads that I didn't realize were leads when I made them. I discussed one example of this in *Ptolus: In the Shadow of the Spire* here, as the unexpected actions of the PCs added two completely new nodes to Act I (even while they were, as noted above, ignoring six other nodes I had planned).

Something similar happened in the *Eternal Lies Remix*, with the players choosing to go to the Severn Valley by following information that wasn't explicitly designed as a lead, but which they nevertheless interpreted as such.

As the GM, you may also find yourself proactively creating new nodes that arise logically out of the evolving circumstances of the game. The Shipwrights' Ball in my *Dragon Heist* campaign is an example of that.

As you're adding these nodes to your campaign, don't forget to link them to other campaign nodes just like you would with any other node. You don't need to worry about making sure to include three clues pointing *to* the new node (you're designing the node because the PCs are heading there; you don't need redundancy to make sure they do the thing they're already doing), but seize the opportunity to add new leads pointing to other nodes. (For example, when adding the *Laboratory of the Beast* to *In the Shadow of the Spire*, I added several new links to Act II of the campaign.)

These connections will often arise naturalistically in any case. If the PCs become interested in a mobster's yacht or lawyer or ex-wife because of their connection to the mobster, it just makes sense that those things could be potentially connected to the mobster's other affairs (i.e., the other nodes connected to the mobster in the campaign).

> *Note: There are plenty of exceptions to prove this "rule." If the PCs go haring off on a wild goose chase, it's perfectly fine for them to discover that it's a dead-end in terms of their wider investigation. (Of course, this doesn't mean that they won't find something* interesting *there. The Manse of the Red Dragon sounds pretty exciting even if it doesn't have any connection to the conspiracy of Elf Lords.)*
>
> *In some cases, you may find that the PCs have blasted open a whole new multi-node section of the campaign. That's great! Just remember not to prep more of those nodes than you immediately need to.*

META-SCENARIOS

I spoke earlier about the two prongs of mystery design: That there are leads which point to other nodes where you can continue investigating things and clues that point to other types of revelations (like the solution to a mystery). Furthermore, the latter can be spread out over the course of an entire scenario, with the PCs slowly accumulating the clues they need to piece things together.

This same principle can hold at the campaign level: You can take a big mystery, break it down into separate revelations, and then spread the clues for those revelations out over the entire campaign (perhaps dropping only one or two in a given scenario).

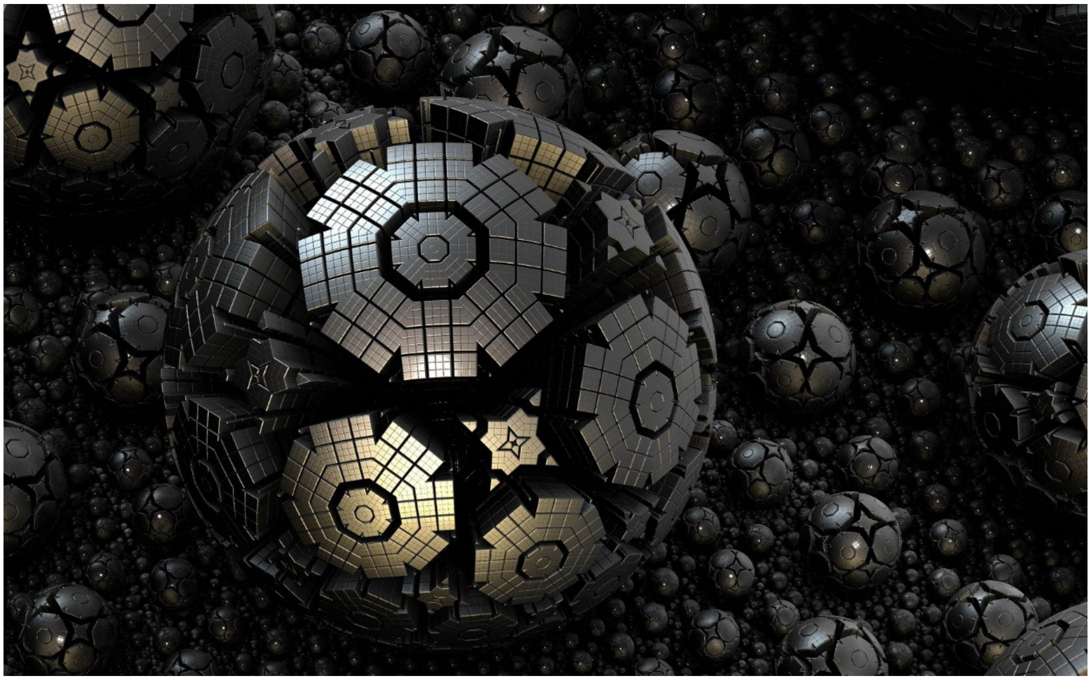
This is a meta-mystery, and it's an example of what I call a **meta-scenario**.

Although the design principles here, like much of what we've been talking about, are the same as they would be for an individual scenario, in some cases a difference in scale *is* a difference of kind. Largely, this is a matter of organization: I usually find that meta-scenarios require their own section of the binder. I find them easiest to both design and manage if I think of them as a distinct entity that's kind of being "draped" over the top of the campaign's primary structure.

Meta-mysteries are not the only sort of campaign-spanning meta-scenario. Other examples might include:

- Collecting the specialized components for performing a ritual.
- Making allies for the Prophecy War.
- Being hunted by an intergalactic cabal of assassins.
- A countdown to the apocalypse as the Stars Come Right.

THE FRACTAL NODE



An idea we've sort of been flirting with here is that node-based design is fundamentally fractal: You can "zoom in" on a node and break it apart into more nodes. Or, conversely, you can "zoom out" of the local collection of nodes you've been adventuring in and discover it's all part of just ONE node in a much larger web.

For example, let's consider the Lytekkas vampire hypercorp:

- At the highest level, we map their activity across seven cities: Chicago, the Old Angeles arcology, Shanghai, St. Petersburg, Reykjavik, Cheltenham, and the Lytekkas-Auberjonais L1 colony.
- Pick one of those and open it up: Now you've got nodes representing all of Lytekkas' major projects in the Chicago metroplex. (These would also have connections to the other cities.)
- One of these involves the development of experimental blood-nannies (probably derived from the vampiric virus to mass-Renfield the population; or maybe an effort to control the negative side-effects of being a vampire).
- Look inside that node and you find the half dozen or so people and facilities involved in the program.
- The PCs identify and target a Lytekkas warehouse. This node is a heist scenario. There are elements of a heist scenario that are node-like, but it's really a different structure, so this is the end of the road.
- … or is it? Part of the heist scenario is the opportunity to secure blueprints of the target. We could resolve that with a skill check, but we could also design it as a micro-node-based scenario.

But you can also flip this whole thing around and see that the Lytekkas hypercorp is just one of the immortal corporations which can track their secret history from the 22nd century back to the Dutch East India Company in the 17th century, with each one of those immortal corporations being a separate node.

This fractal quality can be seen not just in the nodes themselves, but in the connections between those nodes: There's functionally no limit to how many leads can be placed in a single node, nor in how many leads can be created that point *to* a node. There is always an opportunity to increase the complexity and interconnectedness of your node map, particularly if we're talking about an ongoing campaign which is developing over time (and, thus, features connections being formed and broken over time).

At this point the pedants may point out that this isn't *truly* fractal because there is a limit to how far we can take this: At some point we'll end up with a node which is a person; which is a sole entity that cannot be subdivided.

But it's more true than you might think: Even individuals can be reconceptualized as a node-based scenario consisting of their job, connections, family, etc. The question is not *whether* you can do this, but whether it is *interesting* to do so.

Manifesting & Managing Complexity

This fractal nature of node-based scenario design, as we've seen, allows us to manifest an almost limitless amount of complexity. That can be quite daunting.

But it also allows us to manage this complexity. One of the functions of node-based design is specifically to chunk information into manageable nodes so that you don't have to try to juggle or keep the whole thing in your head at once.

If you're getting overwhelmed by trying to handle all seventy nodes of the Lytekkas hypercorp, figure out how to categorize those nodes into manageable groups: By city of operation. Or secret projects. Or corporate division.

The "right" division is going to depend on your own personal preferences and the details of the specific scenario. You're looking for the clear conceptual chunks that will allow you to keep on top of incredibly complicated campaigns, while ideally only needing to think about one chunk of the scenario or campaign at a time. (I talk a bit more about what the manageable limits for this are in *Advanced Node-Based Design – Part 4: The Second Track*.)

For a rough-and-ready example of this, let's consider the Bangkok node in the *Eternal Lies Remix*. In my prep notes, you can see that I identified five nodes:

1. Lowman's Townhouse
2. Phikhat Hwan
3. Ko Kruk Island
4. Sirikhan Estate
5. Savitree Hunts the Investigators

The first two nodes are pretty standard node-based design: They're distinct locations with (at least) three clues pointing at each.

But Ko Kruk Island is then broken into three separate nodes: The island itself, the Sirikhan Estate (a mansion on the island), and Savitree hunting the investigators (on the island).

Why?

To manage the complexity.

While the Sirikhan Estate is located on Ko Kruk Island, if the PCs are exploring the island (Node 3) I don't need to think about every individual room inside the Estate (Node 4). And vice versa: if they're in the mansion, I generally don't need to worry about the whole island. Just like separating the individual rooms of a dungeon into separate keyed entries, separating this information lets me clearly focus on (and find!) what's important RIGHT NOW.

(Speaking of fractal prep, of course, the rooms of the Estate are prepped as a location-crawl. So it's individual rooms within a node (the Estate) within another node (the Island) within another node (Bangkok). You can really see how the prep structure lets me precisely narrow my focus.)

Node 5 was actually a late addition to my prep notes. I was originally trying to include that material in Node 3 — it's stuff that happens on the island, so logically it should be in the "Island" chunk of information.

But the complexity of that particular event sequence was bloating the material to a point where I was finding it difficult to organize and reference it. If this is happening to your prep, it's usually a warning sign that you need to break the material apart into more discrete chunks!

On the flip side, if the players become particularly fascinated by some aspect of the scenario or game world, it becomes relatively trivial for you to zoom in on it and explore it in more detail. Keep this in mind even when running the game: Zoom in on the node and/or add connections to it in response to the PCs' actions. Slap in a simple node-template like the *5-Node Mystery* and you're good to go following the players down their rabbit hole. Who knows where it will take you?

Nodes & the Campaign Status Document

Campaign status documents are something I discuss in more detail in Part 4 of *Smart Prep*. The short version is that the campaign status document collects all of your notes on the current, evolving situation of the game. There are a number of cool ways you can use the campaign status document, but the most pertinent one for our discussion here is the scenario updates: Rather than attaching notes like "they killed all the kobolds in Area 3" or "Benny is angry with them" to a bunch of different scenarios, so that they're scattered hither and yon, you collect all of these notes in the campaign status document for easy reference and upkeep.

The key insight here is that properly organized node-based scenarios make it REALLY easy to assemble and maintain your campaign status document. In the status document for *In the Shadow of the Spire*, for example, I can just place "NOD6" at the top of a section (remember those alphanumeric codes?) and drop stuff relevant to that scenario into that section. When the PCs go to NOD6, I can just flip to that page in the status document and have it available for easy reference.

We've also talked about meta-scenarios. Something I'll do is pull upcoming or pertinent details from a campaign's meta-scenarios and drop them into my campaign status document. When the players in my *Eternal Lies* campaign would go to a new city, for example, I would look at the Act II floating scenes (which operate as a very loose meta-scenario), grab a handful that I thought were likely to be appropriate, and drop a list of them onto the first page of my campaign status document. It gave me a kind of mini-menu that I could quickly consult without trying to process or remember the entirety of the meta-scenario in the middle of a session.

In some cases I'll also drop "active" revelation lists into the campaign status document to keep track of them, but for these I'll usually just mark up the master copy in the scenario itself. (That's what it's there for after all.)

Nodes Aren't Everything

With all of this being said, node-based scenario design is not the be-all and end-all of scenario design. Beyond obvious stuff like dungeon crawls and hexcrawls, I've already discussed lots of other scenario structures here at the Alexandrian. There are also a lot of tools that will be useful to you as a GM in node-based scenarios which aren't nodes themselves.

Let's take another look at my Bangkok prep notes for *Eternal Lies* and break them down. There's five nodes in there, but there's also a bunch of other stuff. What's it all doing in there?

First, you've got the revelation lists and the trigger list for proactive nodes. These are obviously part of node-based scenario design, but they're also basically the table of contents for the scenario. They show the whole scenario structure and can act as a quick reference for your nodes/tools. So I'll usually have this in the first couple pages of my prep notes.

And then we come to the tools. One thing to note is that these are usually designed to be picked up in *combination* with one of the nodes. 95% of the time, this will be stuff that might be used in *multiple* nodes (like an NPC who could show up in multiple locations).

The other thing to note is that there's nothing magical about these particular tools. I don't use them for every scenario I prep and there are a lot of other tools you could potentially include depending on the nature of the scenario and game.

CITY MOOD BEATS: This is a cool tool from the published *Eternal Lies* campaign. These small beats (e.g., "birdsong drifts out of an open apartment window") are designed to be dropped into any scene to provide local color/theme.

NPC NAMES: A list of locally appropriate names to be used when improvising NPCs (which could obviously happen almost anywhere in the scenario).

(These two tools usurp the usual place of honor for the revelation lists and are the first page of my prep notes for the city nodes in *Eternal Lies*. When the PCs went to a city, I would generally grab this sheet out of the binder and place it on the table off to one side for easy use.)

REFERENCE: I use a reference sheet to encode broad background data for the scenario that doesn't belong to a particular node. In this case, it's a timeline for the Emporium of Bangkok Antiquities. (This timeline could come up while questioning Daniel Lowman at his townhouse, questioning Savitree at Ko

Kruk Island, or while looking at research notes found at the Estate. So it's really clear how this has cross-node applicability.)

RESEARCH: In *Trail of Cthulhu* it's not unusual for the PCs to head to the local library or newspaper morgue and do general research on leads they've found. This is material that doesn't really belong to any of the nodes, so it gets split out.

NPCs: At the back of my prep notes I present major NPCs, including the Universal Roleplaying Template and other key information. Some of these NPCs are treated as nodes (and will be found indexed on the revelation list) while others can be used in multiple nodes, but for practical reasons I'll usually include any NPC that takes up a lot of space to declutter the individual nodes. I also just generally find that giving each major NPC a full, dedicated sheet is useful. When roleplaying them, I can just grab and focus on the pertinent sheet. If it's a combat situation, I can also grab the sheets for everyone participating for easier reference.

STAT SHEET FOR BANGKOK: But when a system's stat blocks are short enough (as in *Trail of Cthulhu*), I will also drop all the stat blocks for a scenario onto a single sheet to make running combat SUPER EASY (at least when it comes to referencing stats). Some scenarios might require multiple stat sheets, which I will generally organize according to action group or node.

As I say, these tools are not the be-all or end-all of node-enhancing elements you can include in your scenario and campaign design. But they should give you a pretty good sense of the types of tools I'll develop and use.

NODES & SCENES

I mentioned earlier that you ideally only want to have to focus on one node at a time while running the scenario. In other words, the material should be correctly chunked so that you don't have to juggle material in real time.

While this is mostly true in theory, it's rarely true in practice. There's usually something floating around: a proactive node that's being added to the current scene, a meta-scenario, reinforcements or characters from one node who slipped into another as events played out.

If you *do* need to reach out and grab another tool, the cool thing about node-based scenario design is that it usually bundles those tools up into easy-to-grab packages.

That also makes this a good time to talk about node-based design and another RPG design technique: the scene.

If you look at published adventures, pre-planned scenes are ubiquitous. By their example, a lot of GMs have been trained to prep adventures as a collection or (more often) a sequence of scenes. GMs coming to node-based scenario design for the first time, therefore, are often trying to figure out how the scenes they've learned how to make fit into nodes.

Well, you CAN think of a scene as being a node: Revelations point to the scene, the PCs go to the scene, and then the scene plays out.

Honestly, though, I generally don't prep scenes any more. I find it more efficient and effective to prep the elements that make up scenes and then use those elements to frame scenes in response to the PCs' choices. (For a dedicated discussion of how to effectively frame scenes, check out *The Art of Pacing*.)

Node-based design lends itself well to this approach because, as we've noted, it tends to break stuff up into discrete chunks that are connected (and can be combined) in different ways.

So I don’t prep “the scene with Diego.” I prep Diego and Diego’s mansion, the office where Diego works, etc.

And “the scene with Diego” is a meeting or a tail or an assassination or a heist or an intimidation tactic where they ram his car or a hostage negotiation or all of that. The scene is framed in response to the PCs’ actions; it emerges organically out of play.

Naturalistic Node Design

Once I came to see scenes as an emergent property of game play (rather than something that’s prepared), I began to realize that scenario prep is almost entirely about designing tools. Or, perhaps more accurately, toys. Running an RPG is about picking up these toys and actively playing with them, just like the players are actively playing their PCs.

In creating these tools you’re still thinking about how they’ll be used at the table. But it’s more like designing a hammer (“I know this tool will be useful for hammering nails”) than it is writing an IKEA instruction manual (“here is how the scene with Diego will work”).

At the table, players are going to say things like, “Let’s build a chair!” And the GM will be thinking, “Great! I’ve got a hammer, a bunch of wood, and some… screws? Well, let’s make it work.”

It may only reflect my predilections as a GM, but I’ve found over time that node-based scenario design seems to inherently lend itself to a naturalistic form of prep. I’m not sure if that’s the ethos of active play pushing node-based design in a particular direction or if it’s that node-based design has naturally fostered active play. Or both.

But, in short, the more comfortable I become with node-based scenario design, the more I find that the design arises out of the diegetic organization of the game world itself.

For example, when I’m designing something like the Lytekkas hypercorp, the nodes I design (and how those nodes are organized) overwhelmingly tends to mirror the organization of Lytekkas in the game world. (You can see this in the earlier examples, which were based on things like the geography of the game world or the corporate divisions of Lytekkas.)

When I’m struggling with node-based design, a trick I’ve learned is to pick an NPC and ask, “How would they see this stuff? How would they use it? How would they interact with it?” Whether that’s a hypercorp CEO in an orbital penthouse or a Triad 49er on the rough streets of Hong Kong, the in-game perspective is almost always illuminating.

Of course, this makes sense, doesn’t it?

If node-based scenario design is based on the flow of information – if the structure of a node-based scenario is made up of the paths formed by the information – then it follows logically that node-based scenarios will mirror how that information naturally flows through the game world. And, of course, that flow will be determined by the actual relationships between people, places, organizations, and events in the game world. In how the nodes are related to each other.

Thus, as you design your scenarios, you naturally break the game world apart into discrete chunks. Each chunk (or node) is either a toy or a collection of toys that you can pick up and play with, while the connections between nodes show how those toys are related to other toys, making it easy for them to be endlessly combined and recombined in actual play.

And that’s node-based scenario design.