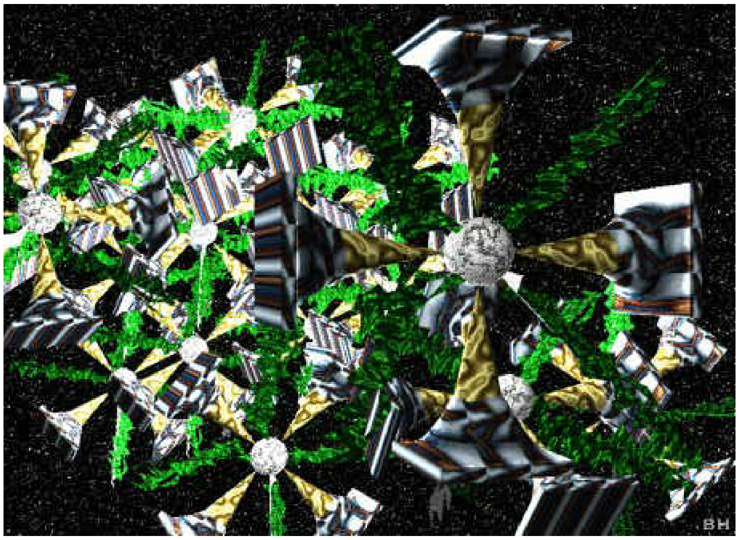
Chaos Processor Layout

Branson Hagerty (Blackjack's Shadowrun Page http://shadowrun.html.com/users/blackjack/) Posted May 06, 1996



The purpose of the Chaos Processor layout is to give a matrix based system a method of processing computations at an extremely high rate of speed without the slow down that results from passing data through various security measures. To achieve this a node structure is established consisting only of SPU modules with its mother CPU. Entry into the system is usually via a single SAN. In order to enter the system a decker must first penetrate several extremely heavy layers of countermeasures but, once in the Chaos system, he is free to go wherever without restriction. When data is usually passed through a system it must be tagged with the various pass codes which allow it access into separate nodes. This slow down results in a considerable longer processing time, especially if the data must pass through several nodes before it is processed. With the chaos system the data moves freely and processing time is decreased. In essence, these are "colorless" nodes.

In order to begin a computation a decker enters the cluster and runs a small utility 'explaining' the operation. If there is any data the system requires initially to begin the computation the decker downloads it now. Instantly the system begins operations to answer whatever question it has been asked. When the computations are done, the decker simply reenters the system and runs a small utility which "collects" the data.

The first thing a foreign decker will notice once entering the cluster is its complete lack of data stores. The Boolean processing used by this type of system is so efficient that all data finishes processing at the same time, it knows which operations will be more difficult and processes them the fastest while easier operations are processed with less "effort". It knows which data it will need before moving on and which data can wait. The result is a system in which virtually every piece of data is active in some way at the same time and everything tends to finish at the same time as well Visually the system appears as a chaotic blizzard of information, which is where the system gets its name. When the computation is complete the blizzard stops and the data falls softly to the floor of the nodes. A decker is usually there within seconds to 'pick it up' and start another computation. A data store would simply be another node which would slow everything down.

Data is either incredibly easy to steal from this system or borderline impossible. If the computation is complete it is a matter of writing a simple "on the fly" utility which downloads the data. If the computation isn't done the decker can write what is called a "butterfly net" utility in order to try to catch the data as it goes by. The problem with this is that if the computation is missing any of its required data it immediately seizes and alerts the system operator that something is wrong. This is one of the only two circumstances under which a Chaos system will communicate with its operator. The other time is when it has completed processing.