The INITONS Catalytic Reflection Between Humanoid DNA and Nero Cell

YAOGUANG LUO, RONGWU LUO
Liu Yang DETA Software Development Limited Company, 410300, China
Changsha Le Hao Fu Chan Hospital,410300, China
E-mail: 313699483@qq.com

ABSTRACT: In order to Emancipate the productive forces, create new productivity, optimize existing production tools to better adapt to the production environment and Better assists Humanoid in where understanding, adaptation and transformation of the environment. We know VPCS [9] architecture is not the end. Absolutely, At least at this paper, will make an implementation in five section: DETA HUMANOID [20] cognition, DETA AOPM Humanoid [20] Cognition Theory, DETA VPCS Medical Business backend logic, DETA IDUQ Catalytic computing and DETA INITONS DECODING [18]. Above all There will spend more and more words in my DETA DNA [14] Law of IDUQ. The Essence of INITONS; DNA [14] theorem: The Essence of DNA [14]; Execution mode of Neuron Calculation; The Essence of Adapting to The Environment; The essence of HUMANOID [20] evolution offspring etc.

KEYWORDS: VPCS [9] AOPM [10], IDUC, Nero, Artificial, Decoder, Medical, PARALLING [19], Computing, Humanoid, ETL, Parser [1], Data Mining [21], INITONS [57]

INTRODUCTION: Recently the real-world first-time human find a solution how to build the magnitude INITONS PDN Life by using computer, now the kernel of this paper will definitely proof that how does this life born and how many scientific tasks we had already finished, finally, also includes the pending tasks for example NUWA Plan, PDE and TVM etc. thus all and all will be implemented in the TEXT section where on the research factors way of how we finding.

TEXT

1. DETA OSS Plan

Since the INITON [57] of the DETA OSS, there has a lot of questions where based on the HUMANOID [20] DNA [14] catalytic computing, for example AI [17], Plan a start as below. Code a problems solution software like a way of YAOGUANG Luo 's cog-style life [54]. Look at this Figure 1, smartly, build basic foundations first, then create more business software where based on this foundation, and finally swap to a HUMANOID [20] model. Many times, hope the model could be an Immortality.

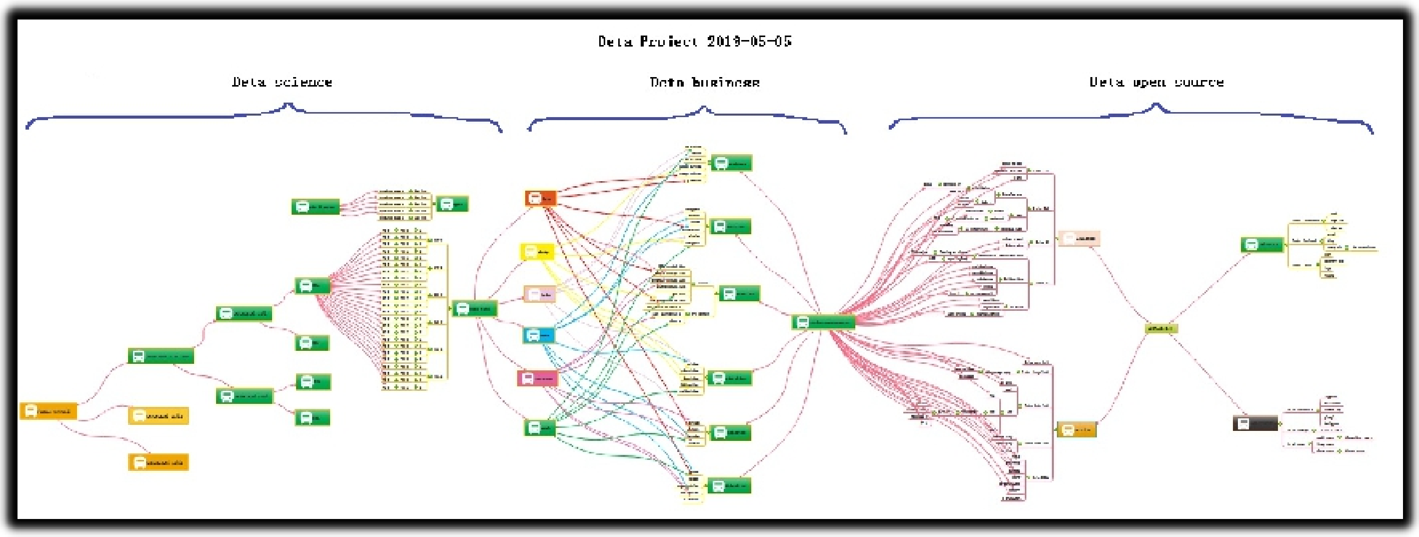


Figure 1 DETA OSS NUWA PLAN

2. DETA AOPM Humanoid [20] Cognition Theory

This process is summarized as the process of analysis, operation, processing and management. The life cycle of software engineering [5] is well explained here. Analysis A, Operation O, Processing P, Management M, use simple words to describe that even if unknown data are collected and analyzed, and then things are operated, the solutions to various difficulties encountered in the operation process are implemented. Finally, maintain and manage these implementation experiences. The back-end computing mode and system life cycle of DETA have gradually condensed from the earliest collection, analysis, operation, sorting, coding, running, debugging and maintenance to the module modes of **Analysis A, Operation O, Processing P and Management M**, such as DETA word segmentation, DETA DNN mind reading, etc. Now ETL [2] of DETA is ready to go in this direction. The author designed a paper last year to describe the application mechanism of AOPM [10] as follows: AOPM [10] Open Source System on SDLC [31] Theory

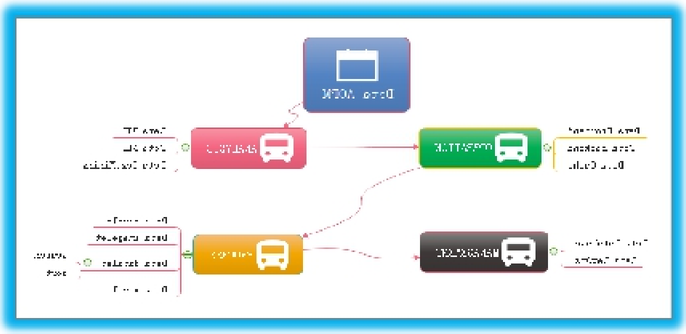


Figure 1-1 AOPM [10] Applications with SDLC [31]

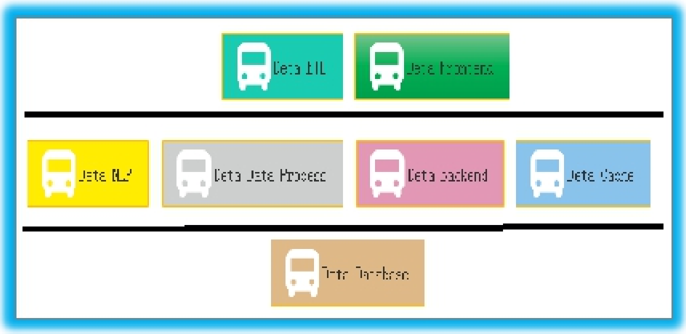


Figure 1-2 Sections of DETA Projects Group

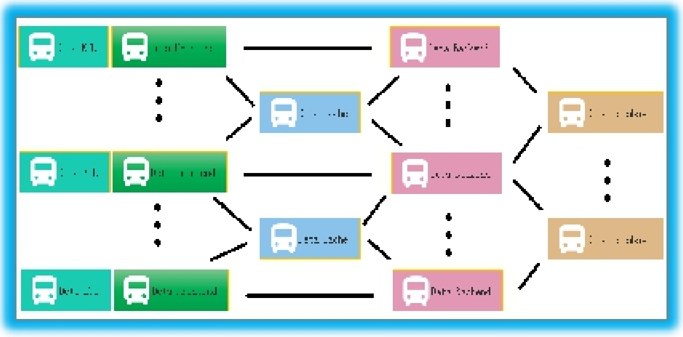


Figure 1-3 DETA WEB Projects System

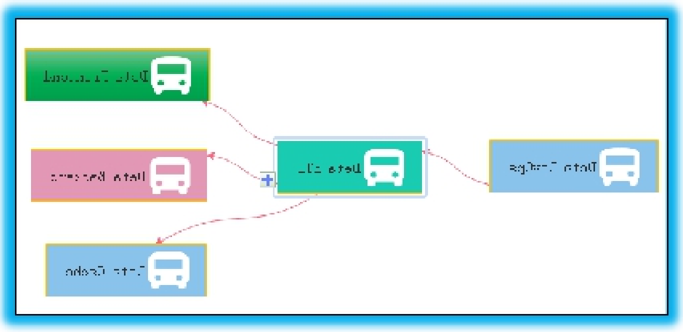


Figure 1-4 DETA DevOps Projects System

Evolutions, absolutely: with connections, what support me the necessary energy for continuing development on my projects. What means connection? Be an internal union bridge between my projects. For example, DETA NLP and DETA ETL [2], they both have the same attributes such as AI [17], Analysis and Data etc. With this connection, my tasks became more dynamically. DETA projects totally can be separated into three dimensions. Frontend Backend and Storage, as the Figure 1-2, the connection between DETA projects is WEB AI [17], now is a Bazaar requirement, but will easy to make estimation of its future, toward to Cathedral. At figure 1-2, DETA open source main based on AI [17] domain, it already formed as an ecology system, go ahead to the application, thanks. Applications, one question is my friend asked me why does DETA support the e-commence logic? Definitely! Please see the Figure 1-3, this is a classic horizontal deployment sample of the real word. Alibaba, Amazon, eBay and JD etc., all based on this technology, instead of Spring, DETA can be the next generation of technology. At Figure 1-4 is a real sample for web DEVOPS by using DETA Open Source.

3. DETA VPCS Business Backend Logic Applications

In the whole year of 2019, the VPCS [9] back-end engine gradually formed some standardized functions and papers, which were applied to the front-end, back-end, cache, database and other subsystems of DETA. My evaluation of them is that they are lightweight and extensible. VPCS [9] is gradually integrated into the works of YANGLIAOJING [11] and HUARUIJI [12]. Of course, there are many shortcomings, the biggest one is that they do not repair themselves. Although I designed the sleeper and hall keeper mechanisms, these mechanisms are only the corresponding business logic units that I complete through decision trees, not HUMANOID [20] evolutionary thinking. At least, I don't think they are HUMANOID [20] intelligence. To be precise, at present, they are only artificial intelligence, a kind of artificial intelligence logic corresponding to AOPM [10] and VPCS [9], but not the HUMANOID [20] evolutionary intelligence logic that I want. Started to explore HUMANOID [20] computing again. About the application principle description of VPCS [9], the author designed a paper as below: VPCS [9] Backend Theory and Its Application. Finding a new method of how to integrate the sets about the micro satellites service in the same server, and make them small, lightly and faster for the commence service, now become a fatal topic. Which can be a pretty warm-up for where makes an explanation for VPCS [9]. The VPCS [9] model, only includes four aspects. **V Vision, P Process, C Controller, S Sets**, and those factors makes an interaction in the sleeper containers. Let talk about the definition of the sleepers. From the software engineering [5] domain, the sleepers are more likely an identified thread person. Who can make a lot of fantasy dream in a Hall, what means a dream? Dream is a requirement what the consumer really needs to finished. But here the dream can be separated out more tasks, those tasks will register the ID in the Pillow, so that the sleeper hugs the pillow then goes into the hall and makes a dream. Got an idea? Cool. So, what does the sleeper does in a hall? The answer is to make all kinds of dreams. For example if they want to build the web service to get rest call, and return the JSON feedbacks, only need to do like a way: Firth, build rest call path in the controller; Second: register the call requirements as a dream; Third, build the sets of the dream in the pillow, Fourth hire a sleeper to hug this pillow, and go to the hall to make a dream process. At last but not the least: return the dream goods. Any sense? Cool! For this unique instance, you will know that the sleeper was more like a Socket [3], and the hall more like a thread pool, the pillows like the single vision instance, and the sets like a vision storage, the controller and the process those two sections is a common way of the factory model. The steps landscape of the sleeper who makes a dream as below as Figure 2-2.

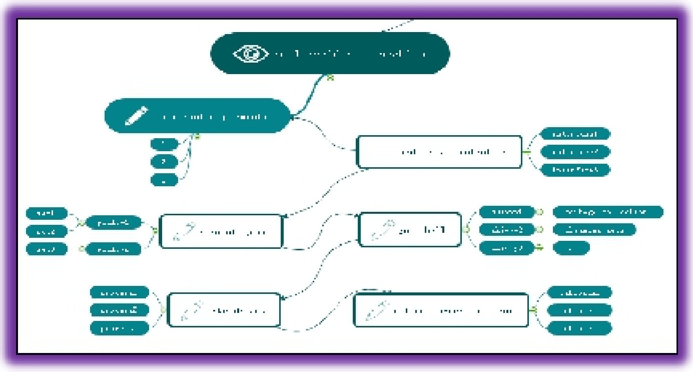


Figure 2-2 VPCS BACKEND MODEL

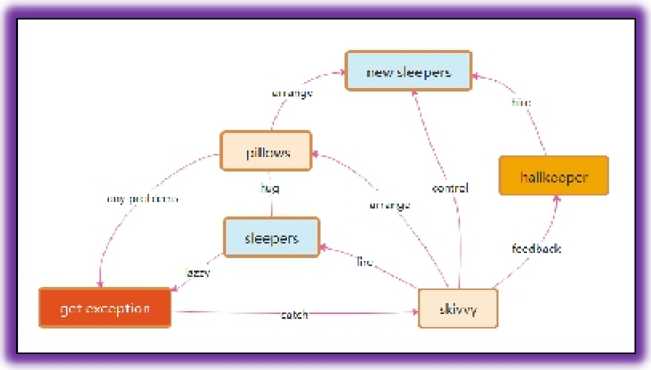


Figure 2-6-2-1 VPCS kernel

Many of these software developers also asked how and why VPCS [9] forks out the LAZZY sleepers excluding their sets. Arthur answered because of the pillows. When the sleepers be hired from the hall keeper, they will get an independently pillows such like static functions. So that sleeper only has their own identify attributes and unique information as the singe instance class. Once they got theirs working paper out, the pillows they used will be arranged to the new fresh sleeper, this theory keeps safe, quality and quantity. Likely Figure 2.6.2.1 VPCS **[9]** kernel.

4. DETA AOPM VPCS IDUQ Catalytic Computing Development

R&D is not successful every time. In the process of butterfly calculation optimization of Fast Fourier, I coded the features of discrete DCT with in complex numbers, which took me one month, but failed. But was excited when I saw the 10th generation of single machine random double with a sorting speed of 12 million arrays per second of quick sorting. Right, and thinning logic is an important way of human thinking. Here, the author designed an argumentation paper when designing fast word segmentation and extremely fast peak filtering catalytic sorting, as follows: Theory on YAOGUANG's Array Split Peak Defect [55]. See Figure 3-2-5 True Top Sort [4][58] Instances, let's show the array data process [58] algorithms here,

```
private int partition(int[] a, int lp, int rp) {
        int x= a[lp]< a[rp]? a[lp]: a[rp]; //reduce the compute values, reduce the recursion peak
        int lp1= lp;
        while(lp1++ < rp){// reduce the condition differential check, reduce the recursion loops
                while(!(a[lp1++]> x|| lp1> rp)) {}
                while(a[rp-- ]> x){}
                if(-- lp1< ++rp){
                        int temp= a[rp]; a[rp]= a[lp1]; a[lp1]= temp;
                }
        }
        a[lp]= a[rp]; a[rp]= x;
        return rp;
}
```

Figure 3.2.5

In order to demonstrate the importance of detailed logic, I began to integrate this logic concept into my YANGLIAOJING [11] and all my software works. When I saw 13 million high-accuracy words segmentation per second; 6 million mixed phonetic symbols per second; 12 million double arrays sorting per second and other amazing works came out, I began to sigh my own cognition. I unreservedly opened up all these ideas and works, which hope aroused humanoid [20] thinking resonance. At present, the significance of differential catalysis has included seven categories: frequency valve (Von Neumann) differential, discrete logic (De Morgan) differential, high-frequency function degradation, conditional refinement differential, executive mode differential, giant system (Qian XUESEN) module differential, and mathematical differential (NEWTON[38], BLAINEZ), which is the only way to catalyze HUMANOID[20] DNA[14] evolutionary algorithm. And made a Detailed demonstration and summarized in 2019[49][50][51][52][53] What is the final expression of Detailed logic? I have never stopped exploring, and I have always been absolutely focused. From 2018 to 2019, the final expression of logic refinement must not be as simple as AOPM [10] and VPCS [9]. VPCS [9] is just a refinement layer of AOPM [10], so how can VPCS [9]be refined? The first demonstration process is DETA word segmentation. In 2019, continued to refine, optimize and refine the word segmentation, and found an exciting argument. DETA word segmentation function was continuously split rationally. Finally, a pile of simple combination application fragments of addition, deletion and modification were displayed by IOAON. For the most powerful argument, when processing nouns in word segmentation, the final function was formed. Memory takes out 4 words, compares 4 words proverbs, does not? then compare 3 words, does not? then compare 2 words, and does not? then split into single words. This process is summarized in one sentence as a combined decision-making process of **I adding, D deleting, U modifying and Q checking** memory data according to John Von Neumann [26]'s time flow form. IDUC is not only the operation mode of database, but also the operation mode of memory data, and it is absolutely focused continuously. Assuming that IDUC/Q is effective for all data operation modes, assuming it is successful, if it is coded, it is a very strict coding mode of data DNA [14]. Seems began to refine my sorting algorithm, word segmentation algorithm, ETL [2], YANGLIAOJING [11], etc., and found one thing in common. All my works were refined to the rational function level that could be understand, which were small fragments of the combined decision-making process of adding, deleting, modifying and checking linear, multidimensional, database and memory data. These fragments can be coded effectively. With this in mind, where determined a ternary mapping coding mode of DNA [14] to ETL [2] neuron nodes, AOPM [10] -VPCS- IDUC 3D coding mode. 4*4*4 Then each primitive is a 64-bit space, which is the computing primitive where looking for decades.

5. DETA DNA [14] Decoding [18] Research

If the AOPM [10] -VPCS- IDUC 3D computational neuron mapped by DNA [14] IDUC is established, how to decode it? about YANGLIAOJING [11]! It is the only way that at present to construct the system of YANGLIAOJING [11] and demonstrate this idea and technology. It is still the absolute focus of that sentence. At present, what is that DNA [14] peptide group has billions of long, double chains and 24 pairs of chromosomes. there are five primitives of ACGTU. if ACGT can encode human higher intelligence logic, then HUMANOID [20] data DNA [14] with IDUC unit can also write hundreds of thousands of business transaction processing logic of AOPM [10] VPCS. these two logics do not conflict. These two logics do not conflict. Are they one? hoping to find a negative theory to overturn it, but unfortunately, I couldn't find it. So, I followed up and re-examined my soft works, optimized them, and found that once optimized to the edge of rational function to irrational function, they were all linear. Small fragments of the combined decision-making process of adding, deleting, modifying and checking multidimensional, database and memory data. These fragments can be coded effectively. AOPM [10] -VPCS- IDUC seems to explain all the answers I want. In order to overthrow my argument, look for arguments everywhere to attack this argument. First, I found the topic of eternal life. According to AOPM [10] -VPCS- IDUC, IDMC is true. Since it can be perfectly explained, I found the topic of infection in COVID-19, that is, DUOP is true, which is an exciting conclusion! Have been searching for answers to all the problems for decades from **AOPM [10] -VPCS- IDUC**, so started array and link swap [59] mapping and coding as follows:

I-INCREMENT/ ADD	**D**-DECREMENT/FILTER	**U**-UPDATE	**C/Q**-QUERY/CHECK
V-VITIONARY/FEEL	**P/E**-EXECUTE	**C**-CONTROL	**S**-SET/STATIC
A-ANALYSIS	**O**-OPERATION	**P**-PROCESS	**M**-MANAGEMENT

Eternal life PDNS Features

I-INCREMENT/ ADD	**D**-DECREMENT/FILTER	**U**-UPDATE	**C/Q**-QUERY/CHECK
V-VITIONARY/FEEL	**P/E-EXECUTE**	**C-CONTROL**	**S**-SET/STATIC
A-ANALYSIS	**O**-OPERATION	**P**-PROCESS	**M-MANAGEMENT**

Metabolism PDNS Features

I-INCREMENT/ ADD	**D-DECREMENT/FILTER**	**U**-UPDATE	**C/Q**-QUERY/CHECK
V-VITIONARY/FEEL	**P/E**-EXECUTE	**C**-CONTROL	**S-SET/STATIC**
A-ANALYSIS	**O**-OPERATION	**P**-PROCESS	**M-MANAGEMENT**

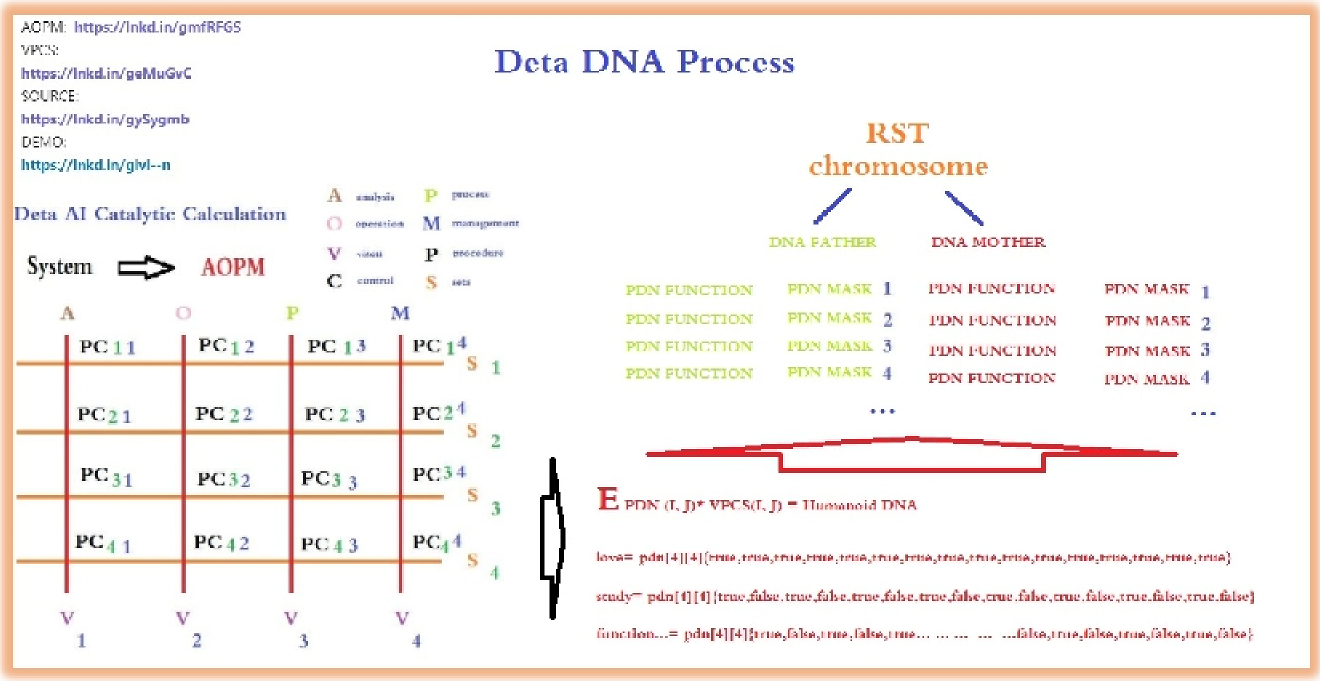
COVID-19 PDNS Features

I-INCREMENT/ ADD	**D-DECREMENT/FILTER**	**U-UPDATE**	**C/Q**-QUERY/CHECK
V-VITIONARY/FEEL	**P/E-EXECUTE**	**C**-CONTROL	**S**-SET/STATIC
A-ANALYSIS	**O-OPERATION**	**P**-PROCESS	**M**-MANAGEMENT

Allergy PDNS Features

I-INCREMENT/ ADD	**D**-DECREMENT/FILTER	**U**-UPDATE	**C/Q-QUERY/CHECK**
V-VITIONARY/FEEL	**P/E**-EXECUTE	**C**-CONTROL	**S**-SET/STATIC
A-ANALYSIS	**O**-OPERATION	**P-PROCESS**	**M**-MANAGEMENT

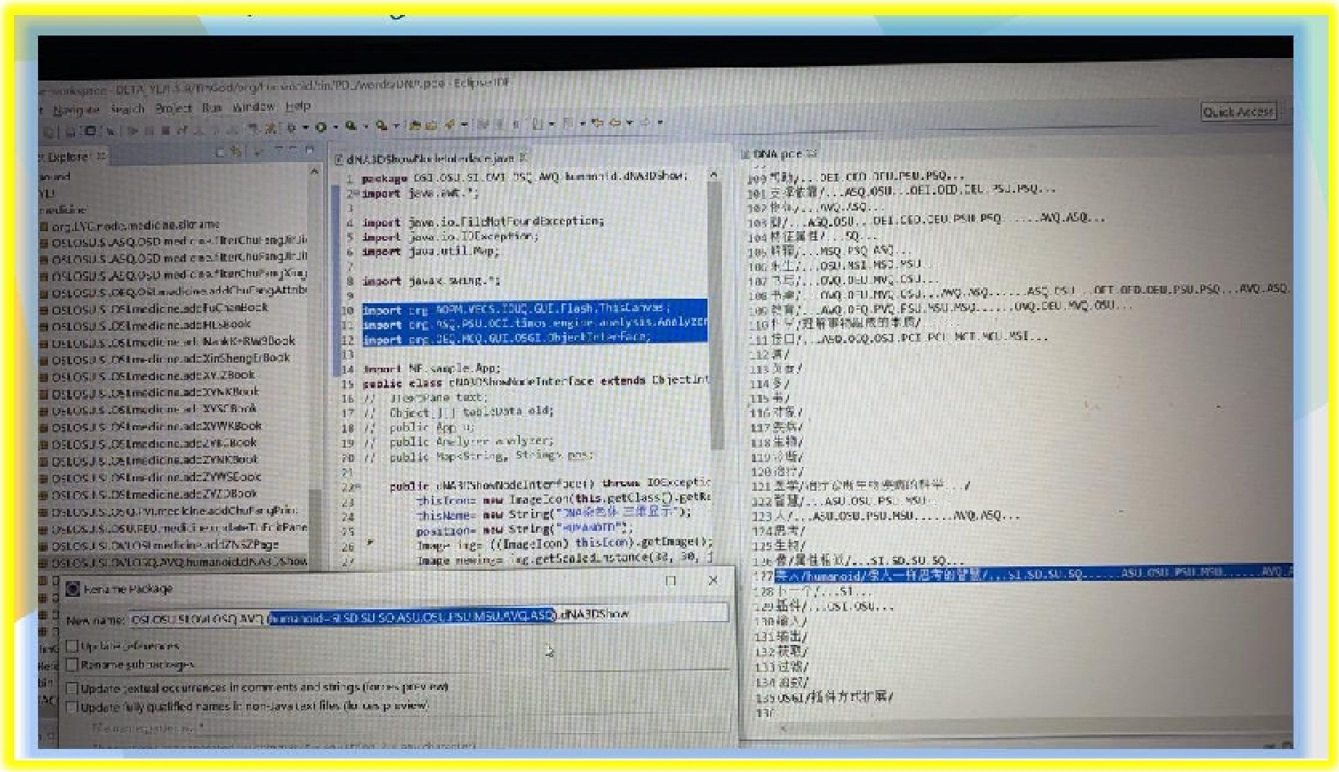
These are all the later stories. The application is too wide. First of all, my ETL [2] began to expand in the three-dimensional direction to better serve medicine. Secondly, virus immunology and immortal virus exploration will never stop. Why ETL [2] is used as the expansion point is inspired by my OSGI [24] paper on October 17, 2013. It is as follows: The Darwin [15]'s Theory of The Artificial Intelligence [56] What about software? The same is true for software. It is particularly important to choose a language that suits your own needs. Secondly, the architecture of the software should have loose coupling, which is similar to OSGI [24] and Felix. The OSGI [24] idea of KNIME [23] is consistent with that of LIFERAY. Although the API design style is different, the effect is very thick. Biology needs Darwin [15]'s thought, and artificial intelligence also exists, which is the basis of demand persistence. This is also the basic condition for my research and development of UNICORN AI [22][17] platform. Now I have enough confidence to continue to focus on the argument of making ETL [2] mapped by my DNA [14] code with evolutionary system reuse the perfect guarantee requirement persistence. On how to use ETL [2] to map the code, I will go back to the previous year again and analyze the design idea of this picture at that time as follows See Figure 6-1, ETL [2] node three-dimensional classification. this vocabulary



AOPM: https://lnkd.in/gmfRFGS
VPCS:
https://lnkd.in/geMuGvC
SOURCE:
https://lnkd.in/gySygmb
DEMO:
https://lnkd.in/givl--n
Deta DNA Process
Deta AI Catalytic Calculation
System
AOPM
A analysis
O operation
V vision
C control
P process
M management
P procedure
S sets
A
O
P
M
PC 1 1
PC 1 2
PC 1 3
PC 1 4
PC 2 1
PC 2 2
PC 2 3
PC 2 4
PC 3 1
PC 3 2
PC 3 3
PC 3 4
PC 4 1
PC 4 2
PC 4 3
PC 4 4
S 1
S 2
S 3
S 4
V 1
V 2
V 3
V 4
RST chromosome
DNA FATHER
DNA MOTHER
PDN FUNCTION
PDN MASK 1
PDN MASK 2
PDN MASK 3
PDN MASK 4
...
E PDN (I, J)* VPCS(I, J) = Humanoid DNA
love= pdn[4][4]{true,true,true,true,true,true,true,true,true,true,true,true,true,true,true,true}
study= pdn[4][4]{true,false,true,false,true,false,true,false,true,false,true,false,true,false,true,false}
function...= pdn[4][4]{true,false,true,false,true... ... ...false,true,false,true,false,true,false}

Figure 6-1

6. IDUC VPCS AOPM [10] 3D Nero Cell and Its Applications



```
package OSI.OSU.SI.OVI.DSQ.AVQ.humanoid.dNA3DShow;
import java.awt.*;

import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Map;

import javax.swing.*;

import org.AOPM.VPCS.IDUQ.GUI.Flash.ThisCanvas;
import org.OSQ.PSU.OCI.timos.engine.analysis.Analyzer
import org.DEQ.MKQ.GUI.OSGI.ObjectInterface;

import ME.sample.App;
public class dNA3DShowNodeInterface extends ObjectInt
//   JTextPane text;
//   Object[][] tableData old;
//   public App u;
//   public Analyzer analyzer;
//   public Map<String, String> pos;

    public dNA3DShowNodeInterface() throws IOExceptio
        thisIcon= new ImageIcon(this.getClass().getRe
        thisName= new String("DNA染色体 三维显示");
        position= new String("+NAXBTM");
        Image img= ((ImageIcon) thisIcon).getImage();
        Image newimg= img.getScaledInstance(30, 30, j
```

Rename Package

New name: OSI.OSU.SI.OVI.OSQ.AVQ.humanoid-$I.SD.SU.SQ.ASU.OSU.PSU.MSU.AVQ.ASQ.dNA3DShow

Update references

Rename subpackages

Update textual occurrences in comments and strings (forces preview)

Update fully qualified names in non-Java text files (forces preview)

Figure 6-2 JAVA > TVM > PDE ENCODING

These are the following words. ETL [2] begins to expand in the direction of 3D. First of all, I want to design the 3 D functional area of neurons based on the DNA [14] mapping of DETA IDUC. This is the real HUMANOID [20] independent thinking way that I can understand. True Instance for <YANGLIAOJING>, See Figure 6-2 The first INDEX [32] application idea of DNA [14] coding manual in human history.

Figure 6-3 DETA Nero Node ETL

‘Org.lyg.node.medcine.addchufangattributeH.jar’ where change to ‘Org.node.a.v.c.u.medcine.addchufangattributeH.jar’ This A.V.C.U. will form a DNA [14] mapping system code for analyzing visual control changes, which is convenient for future evolutionary optimization tests. After that, I will systematically encode these ETL node [2] see Figure 6-3, INDEX [32] mapping sets into DNA [14] INDEX [32] chains for YANGLIAOJING [11]. The ideas given to me in this paper are all creative ideas. Thank you for everything. I can first design AOPM [10] VPCS [9] IDUC INITONS [57] 64-bit+ single chain for the integrity of coding, such as

AVI AVD AVU AVC	API APD APU APC	ACI ACU ACD ACC	ASI ASD ASU ASC
OVI OVD OVU OVC	OPI OPD OPU OPC	OCI OCD OCU OCC	OSI OSD OSU OSC
PVI PVD PVU PVC	PPI PPD PPU PPC	PCI PCD PCU PCC	PSI PSD PSU PSC
MVI MVD MVU MVC	MPI MPD MPU MPC	MCI MCD MCU MCC	MSI MSD MSU MSC

... ... … …

Seems the TVM will translate the Programming File(Java, C, Shell, C++, PYTHON, C#...) Into AOPM-VEC-SIDUQ INITONS PDE File. This INDEX [32] mode, even though it is not the final INDEX [32] of organic DNA [14] of human beings, has become the first mapping execution mode of HUMANOID [20] artificial design representing evolutionarily encoded DNA [14] in chromosome classify prediction[60] model.

```
V    E        V    E        V    E        V    E
  A             O             P            M
C    S        C    S        C    S        C    S

C    S        C    S        C    S        C    S
  A             O             P            M
V    E        V    E        V    E        V    E
```

AOPM VECS INITONS, wisdom dominant chromosome pair determines the way of wisdom association.

```
I   D     I   D     I   D     I   D
  A         O         P         M
U   Q     U   Q     U   Q     U   Q

U   Q     U   Q     U   Q     U   Q
  A         O         P         M
I   D     I   D     I   D     I   D
-----------------------------------------------------------------------------
```

AOPM IDUQ INITONS, wisdom recessive chromosome pair determines the way of wisdom expression.

```
A   O     A   O     A   O     A   O
  V         E         C         S
P   M     P   M     P   M     P   M

P   M     P   M     P   M     P   M
  V         E         C         S
A   O     A   O     A   O     A   O
-----------------------------------------------------------------------------
```

VECS AOPM INITONS, diversity dominant chromosome pairs, determine the diversity of consciousness characteristic.

```
I   D     I   D     I   D     I   D
  V         E         C         S
U   Q     U   Q     U   Q     U   Q

U   Q     U   Q     U   Q     U   Q
  V         E         C         S
I   D     I   D     I   D     I   D
-----------------------------------------------------------------------------
```

VECS IDUQ INITONS, diversified recessive chromosome pairs to determine diversified motion characteristics.

```
A   O     A   O     A   O     A   O
  I         D         U         Q
P   M     P   M     P   M     P   M

P   M     P   M     P   M     P   M
  I         D         U         Q
A   O     A   O     A   O     A   O
-----------------------------------------------------------------------------
```

IDUQ AOPM INITONS, stress dominant chromosome pair to determine the functional aspects of stress.

```
V   E     V   E     V   E     V   E
  I         D         U         Q
C   S     C   S     C   S     C   S

C   S     C   S     C   S     C   S
  I         D         U         Q
V   E     V   E     V   E     V   E
-----------------------------------------------------------------------------
```

IDUQ VECS INITONS, stress recessive chromosome pair to determine the expression object of stress.

CONCLUTION

<u>The Essence of INITONS</u>: An Init AOPM-VECS-IDUQ 3D Aton Sets of Humanoid PDN for DNA Catalytic Computing; DNA [14] theorem: **<u>The Essence of DNA</u>** [14] is a combination Indexing [32] link list of four meta-operations of adding, deleting modifying and Querying data. 10-04-2020 DC; **<u>Execution mode of Neuron Calculation:</u>** a neuron time series calculation chain of specifical function calculation by mapping of DNA[14] coding INDEX[32] reflection; **The Essence of Adapting to The Environment** is that DNA[14] coding Indexes[32] map related neurons compiler link to achieve better addition, deletion, modification and query the environment data; **The Essence of HUMANOID[20] evolution offspring:** the offspring produced by optimizing the hybridization of the same coding logic part in the DNA[14] encoding INDEX[32] chain mapped and retained by the neuron calculation method of data efficient processing. It seems that the topic will never end, so as well boldly put forward new arguments, continuously and tenaciously focus and demonstrate.

REFERENCE

1. YAOGUANG LUO, DETA PARSER, National Copyright Administration of China, CN3951366, (2019).
2. YAOGUANG LUO, DETA ETL, National Copyright Administration of China, CN4240558, (2019).
3. YAOGUANG LUO, DETA Socket DB, National Copyright Administration of China, CN4317518, (2019).
4. DETA TOP SORT, https://github.com/yaoguangluo/sort/blob/master/LYG9DWithDoubleQuickSort4D.java, last accessed 2020/7/14.
5. SOFTWARE ENGINEERING, https://baike.sogou.com/kexue/d10131.htm?ch=fromsearch, last accessed 2020/10/23
6. MVC, https://baike.sogou.com/v25227.htm?fromTitle=MVC, last accessed 2020/10/23.
7. MVP, https://baike.sogou.com/v70887934.htm?fromTitle=MVP%E6%A8%A1%E5%BC%8F, last accessed 2020/10/23.
8. ORACLE, https://baike.sogou.com/v110535.htm?fromTitle=oracle, last accessed 2020/10/23.
9. VPCS, YAOGUANG. Luo
, https://github.com/YAOGUANGluo/DETA_Resource/blob/master/VPCS-Method_V1.1.doc, last accessed 2020/10/23.
10. AOPM, https://github.com/YAOGUANGluo/DETA_Resource/blob/master/AOPM[10]%20System%20On%20VPCS.doc
, last accessed 2020/10/23.
11. YANGLIAOJING, http://tinos.qicp.vip/DETA_HUARUIJI[12].html, last accessed 2020/10/23.
12. HUARUIJI, http://tinos.qicp.vip/DETA_HUARUIJI[12].html, last accessed 2020/10/23.
13. SONAR, https://www.SONAR[12]lint.org/, last accessed 2020/10/23.
14. DNA,
https://baike.baidu.com/item/%E8%84%B1%E6%B0%A7%E6%A0%B8%E7%B3%96%E6%A0%B8%E9%85%B8/78250?fromtitle=DNA[14]&fromid=98123&fr=aladdin, last accessed 2020/10/23.
15. Darwin,
https://baike.baidu.com/item/%E6%9F%A5%E5%B0%94%E6%96%AF%C2%B7%E7%BD%97%E4%BC%AF%E7%89%B9%C2%B7%E8%BE%BE%E5%B0%94%E6%96%87/82699?fromtitle=%E8%BE%BE%E5%B0%94%E6%96%87&fromid=23890,
last accessed 2020/10/23.
16. NERO CELL
, https://baike.baidu.com/item/%E7%A5%9E%E7%BB%8F%E5%85%83/674777?fr=aladdin, last accessed 2020/10/23.
17. AI, https://baike.baidu.com/item/%E4%BA%BA%E5%B7%A5%E6%99%BA%E8%83%BD/9180
, last accessed 2020/10/23.
18. EN-DECODING
, https://baike.baidu.com/item/%E7%BC%96%E7%A0%81%E8%A7%A3%E7%A0%81, last accessed 2020/10/23.
19. PARAL COMPUTING
, https://baike.baidu.com/item/%E5%B9%B6%E8%A1%8C%E8%AE%A1%E7%AE%97, last accessed 2020/10/23.
20. HUMANOID,
https://baike.baidu.com/item/%E4%BB%BF%E7%94%9F%E4%BA%BA/5142593?fr=aladdin, last accessed 2020/10/23.
21. Data Mining, https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%8C%96%E6%8E%98/216477,
last accessed 2020/10/23.
22. UNICORN AI, https://github.com/YAOGUANGluo/ETL_Unicorn, last accessed 2020/10/23.
23. KNIME,
https://www.baidu.com/link?url=g_8i8yfDrvNMqYfN6A9z_XoIc49s8yqzHgpYn-JCBIi&wd=&eqid=88da123b0000be28000000065f7eb5df, last accessed 2020/10/23.

24. OSGI, https://www.oschina.net/p/OSGI[24]?hmsr=aladdin1e1, last accessed 2020/10/23.
25. DML,
https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%93%8D%E7%BA%B5%E8%AF%AD%E8%A8%80/10826467?fromtitle=DML&fromid=10035808&fr=aladdin, last accessed 2020/10/23.
26. John Von Neumann,
https://baike.baidu.com/item/%E7%BA%A6%E7%BF%B0%C2%B7%E5%86%AF%C2%B7%E8%AF%BA%E4%BE%9D,
last accessed 2020/10/23.
27. FFT KERNEL,
https://baike.baidu.com/item/%E8%9D%B6%E5%BD%A2%E8%BF%90%E7%AE%97/4756906, last accessed 2020/10/23.
28. THREAD, https://baike.baidu.com/item/%E5%A4%9A%E7%BA%BF%E7%A8%8B, last accessed 2020/10/23.
29. BLUE TOOTH, https://baike.baidu.com/item/%E8%93%9D%E7%89%99, last accessed 2020/10/23.
30. PDN LINK, https://baike.baidu.com/item/%E8%82%BD%E9%93%BE/8625112?fr=aladdin, last accessed 2020/10/23.
31. SDLC: “”, https://en.wikipedia.org/wiki/Systems_development_life_cycle, last accessed 2020/10/23.
32. INDEX, https://baike.sogou.com/v65431335.htm?fromTitle=%E7%B4%A2%E5%BC%95, last accessed 2020/10/23.
33. SLANG, https://baike.sogou.com/v18863.htm?fromTitle=%E6%88%90%E8%AF%AD, last accessed 2020/10/23.
34. SCRIPT LANGUAGE,
https://baike.sogou.com/v230334.htm?fromTitle=%E8%84%9A%E6%9C%AC%E8%AF%AD%E8%A8%80,
last accessed 2020/10/23.
35. PROSCRIPTION, https://baike.sogou.com/v5065331.htm?fromTitle=%E5%A4%84%E6%96%B9, last accessed 2020/10/23.
36. XUESEN QIAN,
https://baike.sogou.com/v237588.htm?fromTitle=%E9%92%B1%E5%AD%A6%E6%A3%AE, last accessed 2020/10/23.
37. DIFFERENTAL, https://baike.sogou.com/v1966327.htm?fromTitle=%E5%BE%AE%E5%88%86, last accessed 2020/10/23.
38. NEWTON, https://baike.sogou.com/v88517.htm?fromTitle=%E7%89%9B%E9%A1%BF, last accessed 2020/10/23.
49. Leibniz, https://baike.sogou.com/v10692238.htm?fromTitle=%E5%B8%83%E8%8E%B1%E5%B0%BC%E5%85%B9,
last accessed 2020/10/23.
40. RECURATION,
https://baike.sogou.com/v3195520.htm?fromTitle=%E8%BF%AD%E4%BB%A3, last accessed 2020/10/23.
41. POLYNOMIAL,
https://baike.sogou.com/v445974.htm?fromTitle=%E5%A4%9A%E9%A1%B9%E5%BC%8F,last accessed 2020/10/23.
42. Eric Steven Raymond, OSS Book,《The Cathedral and the Bazaar》, ISBN9781565927247, (1999).
43. DETA parser source from GITEE, https://gitee.com/DETAChina/DETAParser, last accessed 2020/10/23.
44. DETA parser split method record, https://github.com/YAOGUANGluo/DETA_Parser/issues/21 , last accessed 2020/10/23.
DETA parser official demo: http://tinos.qicp.vip/data.html, last accessed 2020/10/23.
45. DETA parser integrated products demo from GITHUB:
https://github.com/YAOGUANGluo/DETA_Medicine,
DETA parser integrated products demo from GITEE: https://gitee.com/DETAChina/DETA_Medicine[45],
DETA parser integrated products demo download link:
http://tinos.qicp.vip/download/HUARUIJI[12]Tm1.0.3.zip, last accessed 2020/10/23.
46. Reflection on YAOGUANG's Peak Array Split Defect 1.0,
https://github.com/YAOGUANGluo/DETA_Resource/blob/master/Reflection%20on%20YAOGUANG's%20Peak%20Array%20Split%20Defect1.0.pdf,
Quicksort YAOGUANG.LUO 4D source from GITHUB:
https://github.com/YAOGUANGluo/Data_Processor/blob/master/DP/sortProcessor/Quick_LuoYAOGUANG_4D.java,
Quicksort YAOGUANG.LUO 4D source from GITEE:
https://gitee.com/DETAChina/DETA_Data_Processor_Pub/blob/master/DP/sortProcessor/Quick_LuoYAOGUANG_4D.java, last accessed 2020/10/23.
47. UNIX, https://baike.sogou.com/v5436069.htm?fromTitle=UNIX, last accessed 2020/10/23.

48. LINUX, https://baike.sogou.com/v807585.htm?fromTitle=LINUX, last accessed 2020/10/23.
49. Demonstration of differential algorithm of POS water valve with DETA fast segmentation,
https://gitee.com/DETAChina/DETAParser[43]/blob/master/wordSegment/org/tinos/engine/pos/imp/POSControllerImp.java,
last accessed 2020/10/23.
50. Demonstration of the 4th generation example demonstration of filtering sorting algorithm for LUO YAOGUANG s small peak calculation;
https://gitee.com/DETAChina/DataSwap/blob/dceeb0b06f726d640553964058d85b736354ac89/src/org/DETA/tinos/array/LYG4DWithDoubleQuickSort4D.java, last accessed 2020/10/23.
51. Demonstration of the second generation of differential TSP algorithm for LUO YAOGUANG Euler forest business travel ring
https://gitee.com/DETAChina/Data_Prediction/blob/master/src/org/tinos/DETA/tsp/YAOGUANGLuoEulerRingTSP2D.java,
last accessed 2020/10/23.
52. Demonstration of the 7th generation example of Luo Yao s conditional differential sorting algorithm for light image strings;
https://gitee.com/DETAChina/DataSwap/blob/master/src/org/DETA/tinos/string/LYG4DWithChineseMixStringSort7D.java,
last accessed 2020/10/23.
53. Demonstration of PLSQL differential compiler for DETA Socket [3 stream programmable database engine,
https://gitee.com/DETAChina/DETA_PLSQL_DB/blob/master/java/org/lyg/db/plsql/imp/ExecPLSQLImp.java,
last accessed 2020/10/23.
54. YAOGUANG Luo ‘s cog-style life,
https://baike.sogou.com/v185298047.htm?fromTitle=%E7%BD%97%E7%91%B6%E5%85%89, last accessed 2020/03/06.
55. Theory on YAOGUANG's Split Peak Defect,
https://github.com/yaoguangluo/Deta_Resource/blob/master/Theory%20on%20Yaoguang's%20Split%20Peak%20Defect%201.020190908%20FIX.pdf, last accessed 2019/9.
56. The Darwin's Theory of The Artificial Intelligence, https://313699483.qzone.qq.com/, last accessed 2013/10/17.
57. YAOGUANG LUO, RONGWU LUO, The INITONS Catalytic Reflection Between Humanoid DNA and Nero Cell1.2.2, National Copyright Administration of China, CN2020Z11L0333706, (2020).
58. YAOGUANG LUO, Data Processor API, National Copyright Administration of China, CN4584594, (2018).
59. YAOGUANG LUO, Data Swap, National Copyright Administration of China, CN4607950, (2019).
60. YAOGUANG LUO, Data prediction, National Copyright Administration of China, CN5447819, (2020).