International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

Software Transmutations

M. Ashok Kumar

IFET College of engineering, Department of CSE, Gengarampalayam, Villupuram-605 108, Tamil Nadu, India

Abstract: ***Now-a-days platform dependencies and handling with big data are the very big issues in software development. In order to overcome this problem, I have introduced a new concept called Software transmutations. It is a revolutionary idea of advanced artificial intellectual software development (AAISD).***

Keywords: transmutation, DNA computing, machine translation, *AAISD*

1. Introduction

Transmutation refers to the change of one substance into another. In Charles Darwin's great revolutionary invention about the evolution of species, there comes the concept of genetic transmutation of species and explains how species adapted the change. The memory & thinking capacities of living species are getting smarter every second. Therefore, I introduce the new concept of transmutations in the field of software development in which the softwares will adapt the changes with the platform. It may sounds silly and seems to be a science fiction while comparing the biological factors with that of software and technological development. However, it is the powerful solution to solve the great problems in the handling of big data and resolving platform dependencies and creation of complex AI. When a software created for one platform, does not support the other platforms. Hence, Software mutations are the next generation of creating smart softwares.

2. Self-Adaptability

Installing application software differs across various platforms. They depend upon the extensions or format of the software and the programming languages by which that application was being made. For example, if we need to install simple application under multiple platforms like 'Windows', 'LINUX/UNIX' and 'MacOSX' that applications must in '.exe ', '.rpm,' and '.dmg' extensions respectively. Therefore, it is clear that the user has to depend on the format of the application software. In order to overcome such Dependencies software transmutations are the right choice. A software, which is artificially intellectual enough to detect the operating system & gets Trans mutated in such a way that operating system can run that software.

2.1 Web Applications

In order to perform transmutation our first step is to run an application, which runs without any additional software installations other than that of OS.HTML files makes this task simpler. Web applications can also be used instead of html. The html file along with some client/server side scripts are involved in detecting the type of operating system & starts transmutation according to the self-adaptable format for that particular OS. Web applications are the platform independent services available today. The main advantage of the web applications are that it is too responsive to use, that is the User Interface is too good. Whenever a web application runs in a basic mobile or smartphones, the mobile version of the particular application provides a user-friendly environment.

2.2 Working Mechanism

However, while considering the platforms only in the case of desktop versions, there are variety of operating system classifications. Therefore, the First task is empowering the software with Web programming. We can use either client side or server side scripting languages to detect the type of operating system that is currently running in that device. After detecting the platform, the software has to transmute from a variety of source code into an executable software, which is compatible with our Operating System (OS).This provides the independence on the format of the software as a user by getting rid of this advanced artificial intellectual software development. (AAISD)

Client side scripting (JS):

```
var X="anonymous";
if (navigator.appVersion.indexOf("Win")!=-1) X="Windows";
if (navigator.appVersion.indexOf("Mac")!=-1) X="MacOSX";
if (navigator.appVersion.indexOf("X11")!=-1) X="UNIX";
if (navigator.appVersion.indexOf("Linux")!=-1) X="Linux";
  document.write ('Your OS: '+X);
```

Figure 1: OS detecting utility

This client side script uses attribute type for specifying MIME type. **navigator.appVersion** specifies navigator object using appVersion property to determine the operating system of the client machine. **if(OS_Name.indexOf("Win") != -1)** line contains indexOf which refers to method on collection such as dictionary, hash table etc. and when collection doesn't include the Operating System .The program keep compares with collections. As if it does not matches then it returns, particular OS name on the client system. If it did not find any operating system, and then it displays unknown operating system.

2.3 Beta stage of transmutation – Windows

Here is an example of working demo for the desktop version of windows OS using an HTML Application (HTA)

HTA is a Microsoft Windows program consisting of HTML, Dynamic HTML, and one or more scripting languages such as VBScript or Jscript which executes as a fully trusted application as shown in figure: 3.The HTA file can be linked within an HTML file and can be execute from the default browser.

```
<html>
<head>
<HTA:APPLICATION ID="oMyApp"
navigable="yes"
        APPLICATIONNAME="Application Executer"
        BORDER="no"
        CAPTION="no"
        SHOWINTASKBAR="yes"
        SINGLEINSTANCE="yes"
        SYSMENU="yes"
        SCROLL="no"
        WINDOWSTATE="normal"/>
<script type="text/javascript" language="javascript">
        function RunFile() {
        WshShell = new ActiveXObject("WScript.Shell");
        WshShell.Run("c:/windows/softwares/1.exe", 1,
false);
        }
</script>

      <script type="text/javascript"
language="javascript">

var X="Anonymous";
if (navigator.appVersion.indexOf("Win")!=-1)
X="Windows";
if (navigator.appVersion.indexOf("Mac")!=-1)
X="MacOS";
if (navigator.appVersion.indexOf("X11")!=-1)
X="UNIX";
if (navigator.appVersion.indexOf("Linux")!=-1)
X="Linux";

document.write('Your OS: '+X);

if(X=="Windows")
{
RunFile();
}
</script>
<head>
</head>

</html>
```

Figure 2: program to detect the operating system and runs the appropriate software.

Figure 3: HTA file output wizard

2.4 Beta stage of transmutation –LINUX/UNIX

In such a way much similar to windows, the same javascript code can be used to detect the type of OS and access the application softwares using the terminal shell scripts within the browser with the aid of simple html file.

```
if(X=="UNIX")
      {
      RunFile1();
}

else if(X=="LINUX")
      {
      RunFile2();
}
```

Figure 4: JS script auto detect-OS

Here the RunFile1 () and RunFile2 () consists of shell scripts based on the detected os

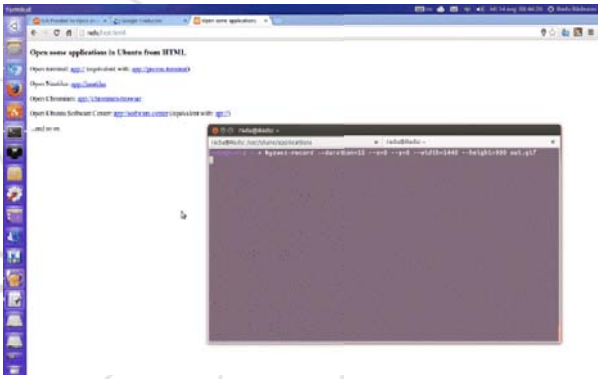

Figure 5: accessing Linux applications using html file

3. Future Work

The Beta model just explains the preceding technique for transmutation. There is still lot of work that has to be done has to be done. When comparing to that of google translate feature our technique differs in the autonomous abilities of the software. In future hardware/software attains the abilities to think and work autonomously using the concept of AAISD.

3.1 Google Translate vs code Transmute

Google Translate helps users to translate text, speech, images, sites and real-time video from one language into another. Now Google can translate more than 103(approx.) languages with accuracy. Google uses statistical machine translation to translate the languages faster.

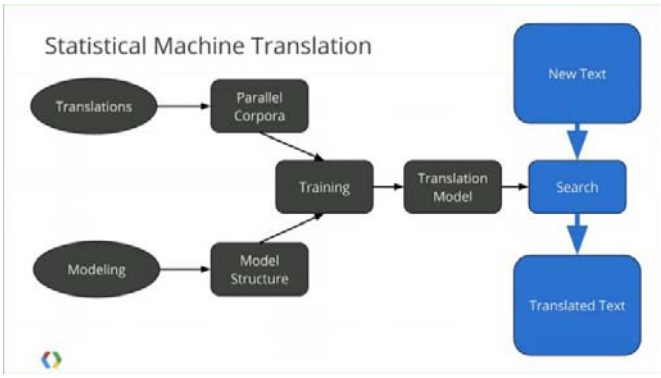

Figure 6: Statistical machine translation

Algorithm used by google to translate the languages. The below diagram depicting the translation of a sentence from German to English using the Segmentation, distortion and translation on the patter basis of the languages.

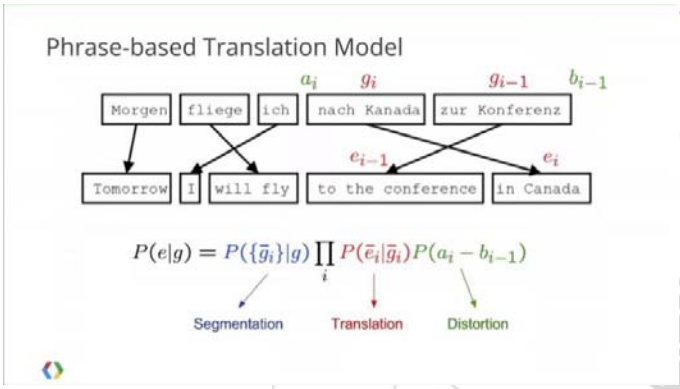

Figure 7: Translation from German-English

Code transmute help users to translate the program from one language into another. It has a mission of providing platform independencies while using the software as a user. It also paves way to the virus free softwares to the end users. It is not a normal source code conversion/translation. It is just an advanced artificial intellectual software, which thinks autonomously. In order to handle the size, computational speed and improve the development of softwares biology has solutions today. AAISD involves the process of Statistical Software transmutation. I am hoping to develop an app for the translation of programming languages into various languages.

But, the Transmutation concept differs from translation. The blue print of software Transmutation is shown in figure: 8

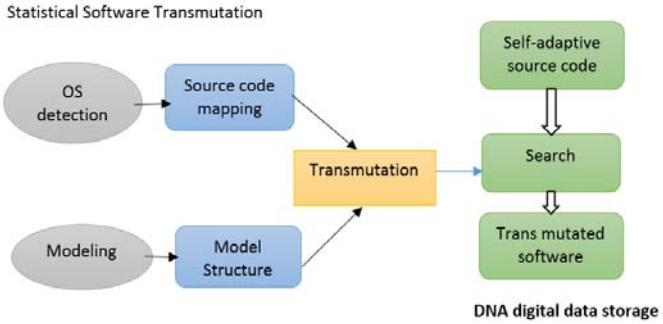

Figure 8: Statistical Software Transmutation

3.2 DNA computing

Leonard Adleman, the inventor of DNA computing, demonstrated the DNA as a form of computation. It solved the 7-point Hamiltonian path problem successfully. A DNA computer with 1 L of fluid containing 6 grams of DNA could have a memory capacity of 3072 exabytes. The data transfer speed would reach 1000 petaFLOPS due to the massive parallelism of the calculations. While today's most powerful computers do not go above a few dozen (33.86 petaFLOPS by Tianhe2 being the current record holder).DNA computing is the key to the smart supercomputers which might be handy, portable and pocket sized which will be going to rule us in the future. The below table shows the necessary bio-software/hardware required for enhancing the transmutation process

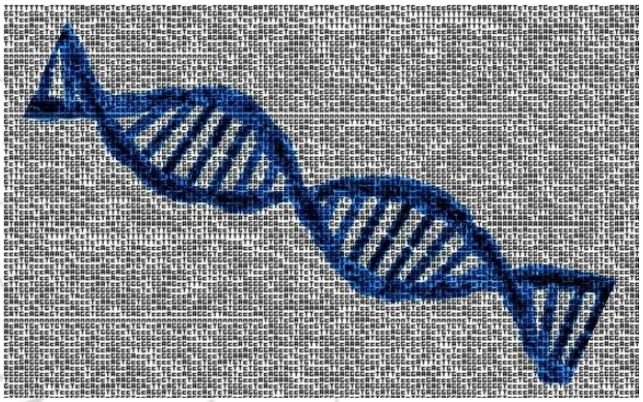

Figure 9: Encoding of Data into DNA

A DNA code is expressed in sequences over the alphabet Q= {A, T, C, G}.
A – Adenine, T – Thymine, C – Cytosine, G – Guanine. These are the basic essential bio-compounds present in the DNA.DNA Code is constructed using complex Hadamard matrices. According to European Bioinformatics Institute, UK 1 test tube of DNA = 1 million CD ROMS. So, it is compact, durable and long lasting storage.

The below fig: 10 representing the DNA encoding and decoding work flow in a parallel execution of multitasks.

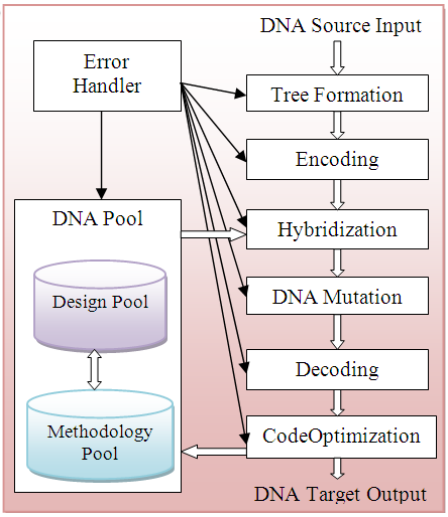

Figure 10: DNA data binding

Table 1: Biological Computers

S.No	Contents	
1	DNA computing	Molecular biology hardware
2	DNA digital data storage	Replacing hard disks
3	DNA code construction	Source code for transmutation

3.3 Challenges

Software transmutations are pretty much difficult than translations. The reason is that the source code has to be encoded in DNA .While considering the object oriented languages which has to be converted into structure oriented languages. In such cases the conversion will works complexly to convert the source codes within the DNA. Converting the compiler dependent languages into interpreter dependent languages adds complexity in a row.

4. Related Work

My project on AAISD is a collaborative project consisting of DNA computing, machine translation, source code conversion. The Georgetown–IBM experiment (Jan 7, 1954) was an influential demonstration of machine translation, which automatically translated more than 60 Russian sentences into English. In January 2013, researchers stored a set of Shakespearean sonnets, a JPEG photo, and an audio file of Martin Luther King's speech, 'I Have a Dream' on DNA digital data storage. Researchers created a transcriptor, which is a biological transistor device, composed of DNA and RNA instead of a semiconducting material such as silicon. Now technology has developed further and entered the fourth industrial revolution with the concept of AI. The speech on the topic 'Future Computing-DNA hard drives' by Nick Goldman explains the detailed overview about Biological Storage devices.

5. Conclusion

It is just the beginning of the biological software revolution. It is clear that the DNA based computations would help us to develop platform independent softwares, thus paving way to AAISD. Web applications with DNA encoding of data will definitely enhance the software creations. I conclude my concept with a citation, "Languages are for communication between god creations where Programming languages are for human creations". So, it is the time for Humans to create Transmutable softwares.

References

[1] http://www.ebi.ac.uk/research/goldman/dna-storage
[2] https://en.wikipedia.org/wiki/DNA_digital_data_storage
[3] https://en.wikipedia.org/wiki/Coding_theory_approaches_to_nucleic_acid_design
[4] https://en.wikipedia.org/wiki/DNA_computing
[5] "Data Storage in DNA" by Siddhant Shrivastava and Rohan Badlani Birla Institute of Technology and Science, Pilani, India International Journal of Electrical Energy, Vol. 2, No. 2, June 2014.
[6] "Applying Software Transformation Techniques to Security Testing" by Thomas Dean and Scott Knight
[7] "Automatically Transforming GNU C Source Code" by Christopher Dahn and Spiros Mancoridis,Drexel University, Philadelphia, PA, USA
[8] 1st International Workshop on Software Evolution Transformations, SET 2004 and Held in conjunction with The 11th Working Conference on Reverse Engineering(WCRE 2004)
[9] "Document-Oriented Source Code Transformation using XML" by Michael L. Collard and Jonathan I. MaleticKent State University Kent Ohio 44242
[10] "High-fidelity C/C++ code transformation" by Daniel Waddington and Bin Yao, United States.

Author Profile

M. Ashok kumar pursuing the third year B.E degree in Computer Science Engineering from Indo French Educational Trust (IFET) college of Engineering in the academic year 2014-2018. Working as a par time blogger in some of the software & programming related websites.