DEVELOP SOFTWARE IDE INCORPORATING WITH ARTIFICIAL INTELLIGENCE

P. C. HARISH PADMANABAN
Department of Software Engineering, PhD Scholar, JJTU University, Rajasthan

ABSTRACT

The software IDE's are most commonly used tool in most of the Software life cycle process.
The demand for open source IDE's are high in the current market and every 5 out of 10 Industries in the Software markets are making use of the IDE's. So the framework is used for the purpose of developing a small piece of code to an complete applications. And it never stops there. It also provides various plug-in and adds on for making the application to more optimistic. Also the same frameworks are used to test the developed application about the functionalities and making sure the application is UN breakable. Sometimes the developed application ended up with the human errors. May be the small error causes the big impact on the signoff phase (SDLC Life cycle). To improve the quality and to develop the application the paper introducing the new IDE which aims to provide the higher impact on developing an application with 98% of optimality. The proposed IDE is named as "D-Intellect". This IDE is one step ahead of normal IDE's/Tools so far developed. The reason behind the IDE is "It will think" with the application developed on the IDE and provide a feedback and suggestions and it not going to stop there. It also provides the complete process and complexity and algorithms which would be helpful for the small scale and large scale enterprise companies globally to build a bug free and environment. The "D-Intellect" stores all the relevant data and compare and analysis process has been carried out to store the data in Knowledge base and it fine tune the algorithms used before and provide the results with the help of predictive analysis.

Keywords: *Artificial Intelligence, SDLC, Data Science, D-Intellect, Machine Learning.*

1. INTRODUCTION

In the near future most of programming should have a basic tool kit or platform to run and validate the performance of the application. The tool to run the code which should be compatible across all the platforms and should have valid recognition. Before even IDE was created the programs and code logics were only running with the help of basic text editors later the compilers and interpreters were developed which makes more comfortable to see the results on the real time. The very first compiler was designed by an American computer scientist Grace Hopper which is **A 0 Programming language** during 1952 with the basic functionality of loading and linking the arithmetic operations. The basic idea for compilers is like compiling the source code "anywhere" and availability across multiple machines. Normally an IDE consisted of text editor, debugger, compiler, loader and linker. The proposed IDE model having the similar basic functionalities of traditional features additionally it is incorporated with the machine learning capability (AI) which helps the code and program highly robust and helps to improve the quality of the application.

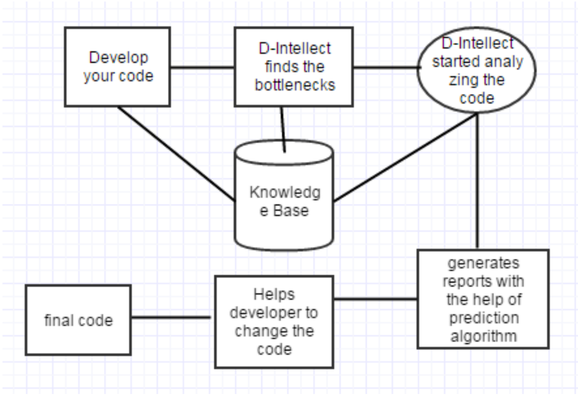
Develop your code
D-Intellect finds the bottlenecks
D-Intellect started analyzing the code
Knowledge Base
generates reports with the help of prediction algorithm
Helps developer to change the code
final code

Fig. 1. Process control flow diagram for Intelligence system in IDE

2. Comparative analysis with different IDE's

There are nearly 50+ IDE's available online in which few of them are free of cost and few of them are licensed and also available for trial pack for few days. All the IDE's are capable of integrating most of the latest third party technologies to make sure the development is in synch with requirements.

Some of the popular IDE's are as follows -

Net beans
Eclipse
IntelliJ
Visual Studio
Power Shell

Each of the IDE's having its own features and characteristics. Open source programming technologies plays a vital role for developing IDE's like java and C++(QT) are few of the examples.

Traditional IDE's vs. D-Intellect

Most of the IDE's are open source and provide wide range of options and features for developing Java applications incorporating with GIT, ANT Maven and other configuration options. Similarly as for D-intellect provide all the features and support cloud technologies like web logic scripting editor and jython(java python) editor .Not just the editors it provides wide range of predictive analysis for the developed code and provide top 3 suggestions which a developer can be followed to change the code highly optimistic. The suggestions are invoked by implementing the Algorithm "**Association rule learning**" which is capable of comparing the variables used in the code with the knowledge base and provides the optimal results. A simple example set for Association **rule learning -**

i/p - {Iphone, powerbank}

o/p -{iphone case cover}

It incorporates artificial intelligence techniques like

supervised learning
Natural sets
Structured prediction
Anomaly Detection

3. Importance of Intelligence system in IDE's

An Intelligence system will be always supportive for keep tracking the code or snippets to ensure it going in a right track. Consider a tutor who is standing behind a student who writing a code (any programming language) in an editor who is responsible for guiding the student through the completion of code. So "**D-Intellect**" works in the same way.

The Intelligence should be incorporated with the business logic to identify the frequent occurrence of code/syntax and should compare the errors with the existing system this can be broadly termed as software Intelligence.

As a humans need a regular learning to keep the brain up to date similarly machine (IDE) also need a regular learning process to keep the knowledge base up to date to provide the valid suggestions. It helps the IDE to support decision making system.

The intelligence system plays an important role for any modern or traditional IDE based systems.

The most important aspects of the intelligence systems are it provides a different view of IDE's with it's at most capabilities. Ultimately it (IDE) should have the capability of change the code which should not break the expected behavior of the development effort. The intelligence systems would not just take the current code and compare the raw data with the KB (knowledge base). It having the ultimate capability of analyzing the code line by line and comparing the patterns and should predict the development behavior.

There is a algorithm called Bayesian inference which helps to identify the behavior of the development activities which is implemented in the proposed system which helps the IDE to predict the mistakes and to improve the productivity of the developers.

The integration part of software intelligence with the IDE is not just a piece of cake but able to understand of lot of internal configurations and learning's required by the IDE itself to evolve by its own with respect to the current code standards and behavior the developers coding skills. Hence with Software Intelligence the IDE undergo a revolution which shows the different perspective of the software with what wedevelop and maintain.

4. UNDERSTANDING "D-INTELLECT"

The "**D-Intellect**" is the proposed Software Intelligence IDE .The word Intellect is a general term which defines "*the power of the mind which having the capability to understand, acquire, learn knowledge*". The D-Intellect provides a solution varies from simple to complex code depends upon the behavior of the developed code. The D-Intellect provides the platform independent feature similar to other popular IDE's available in the software market for example: CodeLite which is one of the popular open source tool specialized for c++, c, PHP and java Script. Currently the proposed system is tested with the platforms - (OSx, Windows and Linux)

Following programming languages, scripts, frameworks, Compilers and technologies are supported by the D-intellect -

1. Java, C/C++

2. PHP, Shell Script

3. Python & R

4. Ruby with (Chef, Rails) frameworks

5. Chef Development Kit integrated feature with trail version

6. WLST (Web logic Scripting tool) compiler

7. Node.js and Angulo js

8. JavaScript

9. GIT and SVN Commit dialog Revamped

10. Integrated Backend Support with open source tools like Oracle SQL, MySQL and No SQL(MongoDB)

11. Provides information for API development with IBM Watson, Microsoft Azure

12. Hadoop implementation impressed with the Apache's library called **mLlib** as part of Apache spark framework.

5. ALGORITHM IMPLEMENTATION

The Proposed system is planned to deliver completely free of cost except few plug-in and third party frameworks which is helpful for the full stack developers for the complete the all the boundaries of application development.

To make the IDE more intelligence. The knowledge base should be more intelligent than the developed code so the implementation of knowledge base is done with more complex algorithms.It would not just search for particular code with the saved data in the KB but it should have the capability of search the syntax and the behavior pattern provided by the developers from the global knowledge base which is shortly coined as Cloud.The basic algorithm implemented to search for a particular pattern is implemented with the help of integrating the few API's named Koders and Google code search which is very helpful for information retrieval and documenting the pattern to store it in the Knowledge Base. So far all the research papers proposed the implementation of API's which never provide the solution for larger structured code , performance and the scalability.

Few Implementation of Prediction systems involved for example : Apriori which analyze the frequent occurrence of the transactional patterns The prediction functionality with example -

for loop	condition	n complexity
for loop	condition	n^2 Complexity
for loop	condition	n(log n) complexity

RESULT PREDICTION -

1. For loop gives 100 % chance of providing a condition

2. For loop with a condition gives 33.33% result of n^2 Complexity and n(logn)Complexity solutions. So improve the code in such a way to n(logn) .

Association rule learning which provides the wide angle of clustering, dimensionality prediction, learning, Anomaly detection , neural networks and reinforcement .Apriori is basically the subset for Association rule mining technique.

This techniques are key recipes for building the more robust and highly intelligence software systems integrating with the IDE's(D-Intellect).

This algorithm is mainly to fetch the interestingness among the patterns invoked previously.Its basic functionality is to mine the particular data with respect to the data mining perspective

Initially It analyze the data with Marlet basket .

Then it generates the rule form.Then It generates the Confidence parameters finally the support.

Apriori Algorithm

```
Apriori$(T, \epsilon)$
    $L_1 \leftarrow \{\text{large } 1 - \text{itemsets}\}$
    $k \leftarrow 2$
    while $L_{k-1} \neq \emptyset$
        $C_k \leftarrow \{a \cup \{b\} \mid a \in L_{k-1} \wedge b \notin a\} - \{c \mid \{s \mid s \subseteq c \wedge |s| = k-1\} \nsubseteq L_{k-1}\}$
        for transactions $t \in T$
            $C_t \leftarrow \{c \mid c \in C_k \wedge c \subseteq t\}$
            for candidates $c \in C_t$
                $count[c] \leftarrow count[c] + 1$
        $L_k \leftarrow \{c \mid c \in C_k \wedge \; count[c] \geq \epsilon\}$
        $k \leftarrow k + 1$
    return $\bigcup_k L_k$
```

k-item set - An item set containing k items

Lk Set of frequent k-item sets (k-item sets with minimum support)

Ck Set of candidate k-item sets (potentially frequent item sets)

Uk Lk Set of generated item sets

Next algorithmic implementation integrated with **D-Intellect** is ***moment correlation coefficient***s which produces the liniear dependence between variables and provide the Boolean results for positive and negative correlations.

Finally the search based engine for updating the knowledge base.

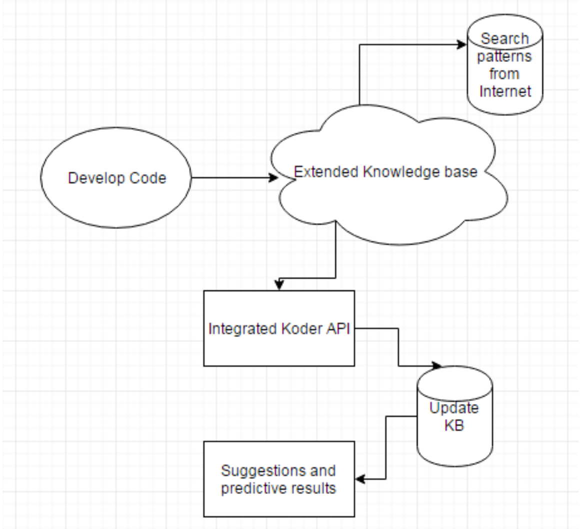

Fig. 2. Search based engine for predicting the values from Internet.

6. CONCLUSION

The current IDE's with eclipse has provided the forum for giving the information on IDE's Enhancements and performance improvement suggestions also there are several webpages where the bugs and issues are provided which has been taken care by the IDE's proprietor. The new intelligence system integrated with D-Intellect will provide surprise to the developers by changing the way of thinking the different IDE systems and processes. So the new features like search based Knowledge evolution and self-decision making process which makes the IDE to be more sophisticated in terms of core developments activities. The Maximum flexi nature of the compiler allows the user to developing the code without any interference of acknowledgements from the IDE.

The proposed system always ensure the existing
Functionality and even if the code is broke it is capable of restoring the old functionality which

In need to restore the old behavior of the applications

REFERENCES

[1] Chris M.Hancock. "Real Time programming and big ideas of computational literacy" , PhD Thesis,MIT,2003.

[2] M Gabel and Z.Su "A study of the uniqueness of the source code" .ACM,2010.

[3] IBM Rational Insight. http://www-01.ibm.com/ software/rational/products/insight/.

[4] Mithun Acharya , Tao Xie , Jian Pei , Jun Xu, Mining API patterns as partial orders from source code: from usage scenarios to specifications, Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering, September 03-07, 2007, Dubrovnik, Croatia

[5] Suresh Thummalapenta , Tao Xie, Parseweb: a programmer assistant for reusing open source code on the web, Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering, November 05-09, 2007, Atlanta, Georgia, USA.

[6] A. M. Turing, "Computing machinery and intelligence," Mind, vol. 49, pp. 433–460, Jan. 01 1950. [2] J. McCarthy, "Programs with common sense," in Proceedings of the Symposium on Mechanisation of Thought Processes, vol. 1. London: Her Majesty's Stationery Office, 1958, pp. 77–84. [3] J. P. Cramer, Almanac of Architecture and Design. Atlanta: Greenway Communications, 2000.

[7] M. Harman, "Software engineering meets evolutionary computation," IEEE Computer, vol. 44, no. 10, pp. 31–39, Oct. 2011.

[8] S. M. Poulding and J. A. Clark, "Efficient software verification: Statistical testing using automated search," IEEE Transactions on Software Engineering, vol. 36, no. 6, pp. 763–777, 2010

[9] W. Weimer, T. V. Nguyen, C. L. Goues, and S. Forrest, "Automatically finding patches using genetic programming," in International Conference on Software Engineering (ICSE 2009), Vancouver, Canada, 2009, pp. 364–374

[10] Y. Matsumoto. A Software Factory: An Overall Approach to Software Production. In P. Freeman ed., Software reusability. IEEE CS Press, 1987

[11] Microsoft. "on{X} - automate your life." www.onx.ms, 2012

[12] Erik Linstead, Sushil Bajracharya, Trung Ngo, Paul Rigor, Cristina Lopes, and Pierre Baldi. Sourcerer: mining and searching internet-scale software repositories. Data Min. Knowl. Discov., 18(2), 2009.

[13] Jinhan Kim, Sanghoon Lee, Seung won Hwang, and Sunghun Kim. Towards an intelligent code search engine. In AAAI Conference on Artificial Intelligence, 2010

[14] Naiyana Sahavechaphan and Kajal Claypool. Xsnippet: Mining for sample code. In International Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), pages 413–430, 2006

[15] D. Rodriguez, I. Herraiz, and R. Harrison, "On software engineering repositories and their open problems," in The International Workshop on Realizing AI Synergies in Software Engineering (RAISE'12), Zurich, Switzerland, 2012