

Improved Entity based Object Oriented Framework for IDE

M.Anil Kumar¹, A.Durga Praveen Kumar², V.Tata Rao³

^{1,2,3}Assistant Professor

^{1,2,3}Raghu Engineering College, dakamarri, Visakhapatnam, A.P

Abstract: Improvement of object oriented concepts in integrated development environment (IDE) is an interesting research issue in the field of object oriented technology in computer science. Every framework has its own advantages and disadvantages. Ability to generate automated data transfer objects and data access objects is a challenging issue for framework. Most of the frameworks show static and dynamic visual representation but they have certain limitations to use object oriented principals. In this paper we propose efficient features to integrate to the IDE which satisfies developer requirements.

I.INTRODUCTION

Object Oriented Programming based on concepts of objects. It contains the data in the form of fields such as attributes, codes and in the form of procedures. A feature of objects is that an object's procedures can access and often modify the data fields of the object with which they are associated. Object-oriented Programming uses objects, but not all of the associated techniques and structures are supported directly in languages that claim to support OOP. The features listed below are, however, common among languages considered strongly class- and object-oriented [1][2].

In spite of the way that educational programming conditions have been accessible for quite a long while, some crucial ideas of object oriented programming and plan, for example, encapsulation, interfaces, polymorphism, and legacy, are not tended to by existing apparatuses to a full degree[3]. The recent studies explained the novel collaboration similitudes epitomized in instrument bolster that, instead of existing devices, makes conceivable to delineate such object oriented programming ideas with top of the line portrayals. A device alluded to as AguiarJ, which is being utilized at our foundation for as far back as three years, encapsulates novel association representations for tending to the previously mentioned ideas in a slow and far reaching way.

We clarify the imaginative device highlights utilizing a running case including the area of picture control [4].

While object-orientation has been broadly acknowledged as an essential programming worldview, showing object-orientation stays troublesome. Encounter reports recommend that a few issues can be maintained a strategic distance from by showing object-orientation as the primary dialect in an early on course. These representations help understudies better see how language structure means the development and utilization of occasions of classes, how each line of code impacts the programming condition, and how to successfully use programming aptitudes. The three essential UML diagrams utilized among these instruments are class diagrams, object diagrams, and arrangement diagrams. Class diagrams outwardly exhibit each class of a program and how classes are interrelated. Object diagrams take into account a more particular understanding of a specific program, indicating occasions of each class that have been produced. Grouping diagrams demonstrate the introduction of s, cases of classes, and strategy calls and in addition the lifetime of each of these components [5][6].

II. RELATED WORK

Static visualizations are those that don't change with time. Since the learner does not connect with such a perception, these assume to some degree restricted part. This is particularly valid if the learner enters with misguided judgments of an idea[7]. A picture of a chart or a graph or any type of static representation won't just not be a persuading media for indicating inconsistencies in understudies convictions, yet will likewise be translated inside the setting of their confusion. This does not mean, in any case, that static visualizations can't assume a part in training. Essentially those visualizations are more suited for

building connections between ideas than actuating disequilibrium inside the misguided judgment of the understudy.

We can draw an analogy to another domain of knowledge. These analogies use to explain that there are various notations of the method or topic. In that visualizations used to create the arcs in the data network of the user. Initially we need to learn the word problems and have difficulty constructing a semantic model that represents in the form of text. Because of this they have difficulty constructing the correct number sentence to solve the problem. The visualization below draws on the students concept of sets and counting. In doing so it makes direction between the word problem and knowledge they have of the world[8].

Serve as an intermediate representation in the case that the concept to be learned is too distant from the knowledge base of the learner. The in the above visualization the intermediate set representation is not meant to be a lasting fixture on the path from word problem to number sentence. It serves as an intermediate representation while the direct links are being formed. By using the set representation, however, the set/counting part of the network has become more integrated into the knowledge base. This integration can again be utilized in teaching other material[9].

The following is a couple of visualizations for a Stirling heat cycle. The picture on the left is a temperature (T) - entropy(S) graph. The other is a pressure (P) - volume (V) chart. Joined these outlines demonstrate the connections between weight, volume, temperature and entropy as the burn proceeds through the four stable conditions of the cycle[10].

III. PROPOSED WORK

We propose an empirical model of object oriented concepts improvement in IDE . Most of the IDEs provides static and dynamic representation of the classes and it helps the user to identify the bugs while running the application. You can a set a break point at line of code, it stops the execution when it comes the execution of that point and we can see both static and dynamic representation side by side, it helps the programmer to identify the issue or bug easily and reduces the development effort.

IntelliSense is the general term for a number of features: List Members, Parameter Info, Quick Info, and Complete Word. These features help you to learn more about the code you are using, keep track of the parameters you are typing, and add calls to properties and methods with only a few keystrokes. Many aspects of IntelliSense are language-specific. For more information about IntelliSense for different languages, see the topics listed under See Also.

IntelliSense helps to get parameter information like name, number and type of parmeters needed by a method and generic type of the attribute or parameters. The attribute which is highlighted indicates the next parameter that is required as you type the function. For overloaded functions, you can use the UP and DOWN arrow keys to view alternative parameter information for the function overloads. Gives complete information of the method or documentation of the methods, classes and interfaces.

Traditional IDEs can attach language specific jar file or dll files or other assemblies. If framework supports cross language build integration then it would be master piece to the programmers. They need not worry about language builds.

```
string s = "hello";  
bool b = s.EndsWith("o", |)...
```

▲ 2 of 3 ▼ **bool string.EndsWith(string value, StringComparison comparisonType)**
Determines whether the end of this string instance matches the specified string when compared using the specified comparison option.
comparisonType: One of the enumeration values that determines how this string and value are compared.

```

InitializedComponent();
string s = "hello";
bool b = s.EndsWith(

```

bool string.EndsWith(string value) (+ 2 overloads)
 Determines whether the end of this string instance matches the specified string.
 No overload for method 'EndsWith' takes 0 arguments

Object framework:

Traditional Entity frameworks can generate dynamic visual representation of the classes by converting the table to model to data transfer object but it does not automatically do the mapping with mapping or join columns. Most of the development effort for programmers is creating models or data transfer objects for respective tables, data access or

manipulation methods to perform read, insert, update and delete operations. We propose a feature to create a data transfer object and data manipulation automatically when table created along with its relationship exists with key columns. It gives complete object oriented structure and makes the implementation simpler.

| Patient |
|-------------|
| PatientId |
| Gender |
| DOB |
| InsuranceId |

| Insurance |
|-------------------|
| InsuranceId |
| MasterInsuranceId |
| PlanId |
| EffectiveDate |
| Enddate |

| InsuranceMaster |
|-------------------|
| MasterInsuranceId |
| Code |
| AddressLine1 |
| Addressline2 |
| City |
| State |
| Zipcode |

```

Class Patient
{
    Public String PatientId { get; set; }
    Public String Gender { get; set; }
    Public String DOB { get; set; }
    Public String InsuranceId { get; set; }
}

```

```

Class Patient
{
    Public String PatientId { get; set; }
    Public String Gender { get; set; }
    Public String DOB { get; set; }
    Public String InsuranceId { get; set; }
    List<Insurance> InsuranceList: New List<Insurance>(){get; set;}
}

```

Most of the IDEs supports entity conversion from tables but cannot establish collection relation ,if we can make optional ,it will be very beneficial to the users because ,in real time scenario entity may have many number of collections ,t is not automatically converted in entity framework . For Example ,in above example patient may have multiple insurances and insurance id gathered from master insurance.

IV.CONCLUSION

We have been concluding our current research work with efficient features like intellisense, object or entity framework which converts the tables to data transfer objects along with cild relation inentity model, so developer effort can be reduced

and simplifies the crud operations. IDE can be optimal when it can show complete object oriented structure of its own. Our proosed research work gives more efficient performance than traditional model.

REFERENCES

1. Joseph Bergin , Eugene Wallingford , Michael Caspersen , Michael Goldweber , Michael Kolling, Teaching polymorphism early, Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education, June 27-29, 2005, Caparica, Portugal [doi>10.1145/1067445.1067541].
2. Steve Cooper , Steve Cunningham, Teaching computer science in context, ACM Inroads, v.1 n.1, March 2010 [doi>10.1145/1721933.1721934]
3. Erich Gamma , Richard Helm , Ralph Johnson , John Vlissides, Design patterns: elements of reusable object-

- oriented software, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1995
4. Mark Guzdial, A media computation course for non-majors, Proceedings of the 8th annual conference on Innovation and technology in computer science education, June 30-July 02, 2003, Thessaloniki, Greece
 5. J. Yang, Y. Lee and D. Hicks, "Synchronized Static and Dynamic Visualization in a Web-Based Programming Environment," IEEE International Conference on Program Comprehension (ICPC), May 16- 17, 2016
 6. Michael Kolling, "The design of an object-oriented environment and language for teaching," PhD Dissertation, University of Sydney, 1999.
 7. Paul Gestwicki, Bharat Jayaraman, "Methodology and architecture of JIVE," Proceedings of the 2005 CM symposium on Software visualization, May 14-15, 2005, St. Louis, Missouri.
 8. T. Dean Hendrix, James H. Cross, II, Larry A. Barowski, "An extensible framework for providing dynamic data structure visualizations in a lightweight IDE," Proceedings of the 35th SIGCSE technical symposium on Computer science education, March 03-07, 2004, Norfolk, Virginia, USA.
 9. 1. Deborah J. Armstrong, The quarks of object-oriented development, Communications of the ACM, v.49 n.2, p.123-128, February 2006 [doi>10.1145/1113034.1113040]

BIOGRAPHIES



M. Anil Kumar is working as assistant professor in raghu engineering college dakamarri, in bheemuni-patnammandal, visakhapatnam.



A. Durga Praveen Kumar is working as assistant professor in raghu engineering college dakamarri, in bheemuni-patnammandal, visakhapatnam.



V. Tata Rao is working as assistant professor in raghu engineering college dakamarri, in bheemuni-patnammandal, visakhapatnam.