ISSN No. 0976-5697

Volume 8, No. 5, May – June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Research Paper on Implementation of OCL Constraints in JAVA

Er. Shivani Gupta
Research Scholar
Dept. of Computer Eng.
Punjabi University, Patiala, India

Dr. Dhavleesh Rattan
Assistant Professor
Dept. of Computer Eng.
Punjabi University, Patiala, India

Abstract: This paper provides an introduction to the UML in the field of software engineering. The first part gives information about the UML class diagrams. Second part of this paper provides an introduction to the formal specification of UML diagrams using OCL. It also provides a detail about why formalization of UML diagrams is important. It also gives introduction about how constraints are applied to the UML diagrams using OCL. It provides knowledge about different kinds of constraints which any applied on UML class diagram. Last part of this paper shows the source code implementation in java of Bank account example of OCL constraints.

Keywords: UML, class diagram, formalization, OCL, pre-condition, post-condition.

I. INTRODUCTION

Unified Modeling Language is a modeling language in the field of software engineering. UML is the de-facto standard for system modeling [1].UML provides a standard way to visualize the design of a software system .UML was adopted as a standard by object Management group (OMG) in 1997 [1].UML was approved as an ISO standard in 2005.UML is the process of understanding visual model of a software system.UML is not a programming language. UML diagrams contain several different types of diagrams:-class diagrams, use case diagrams, state diagrams, interaction diagrams, activity diagrams and deployment diagrams. In this paper we deal with the class diagrams [12].UML model is used for the description of the system.UML diagrams captures decisions and understanding about how the software system is to be constructed .UML captures whole information about the system.UML is not intended for theorem proving.UML was developed to simplify the object oriented development methods.

A. UML Class Diagram

Class diagram is a static diagram. It shows the static view of the system. Class diagram describes the attributes and operations of the class. Class diagram is a collection of classes, interfaces, associations, collaborations and constraints. Class diagram is also known as structural diagram. Class diagrams are mostly used in the modeling of object oriented system because they are the only UML diagrams which can be mapped directly with the object oriented languages. Overall quality and efficiency of the software design is enhanced by using UML as the standard modeling language for the design of the software [14]. UML class diagram acts as the communication medium between the software specification at user side and software realization at developer side [10].Class diagram is used for the analysis and design of the static view of an application and to describe the responsibilities of the system .Class diagrams describe the functionalities performed by the system.

In class diagram, classes are shown by boxes that contain three compartments:-

- In First compartment exists the name of the class.
- In Second compartment exists the attributes of the class.
- In Third compartment exists the operations of the class.

B. Formalization of UML Class Diagram

First step in the development of the software system is the formal specification. Formal specification is then followed by verification, refinement and then implementation. The aim of formal specification is to provide a precise and unambiguous description of the system. Through formal specification designer can verify important properties of the system, can solve any problem and detect any error before starting the development of software system. Formal specification of the system is required so that requirements of the system can be easily understood [5].It also provides a way of communication between system designer and the persons who are involved in the development of software system. It acts as a contract between the system designer and the client for which the software system is to be built. UML is easy to use formal methods are very difficult to apply in real practice [9]. It provides a blueprint of how the system is to be built. Formal specification use mathematical notations to describe in a precise way the properties which an information system must have. Formal specification only describes what a system must have but without describing how it is to be done.

C. Object constraint language

Object constraint language was developed in 1995.OCL is designed in such a way that it is formal and simple. OCL is used to make model accurate, consistent and more precise [2]. The syntax of OCL is very simple and can be learned easily by anybody who is familiar with programming language [2].OCL adds precision to the UML diagram. OCL supports the expression of invariants, preconditions, postconditions so as to allow modeler to outline the unique constraints on the behavior of the model [6].OCL adds details to UML model. Within UML, OCL now becomes a standard to add additional information to the UML model. In UML version 1.1, OCL adds only constraints to the UML model .but in UML version 2 including constraints some other additional information was also included. Defining queries, referencing values, stating conditions or business rules were all included by OCL [3]. But OCL is still an immature language [4].Recently a new version of OCL version 2.0 has been defined in the object constraint language specification.OCL is very simple formal specification language. OCL is used in combination with UML so as to overcome the limitations of UML so that detailed aspects of the system design can be accurately specified [3].OCL includes very simple concepts which can be understood easily.OCL is very compact but powerful formal specification language. OCL provides the possibility to express business rules in formal and unambiguous manner .OCL gives the feel that it is just like other object-oriented programming language.OCL does not include any mathematical symbols or advanced complex concepts. It includes syntax that can be very easily understood.OCL is a pure specification language, an OCL expression is without any side effects [3]. When OCL expression is evaluated, value is returned.OCL expression does not change anything in the model. Combining UML diagram with OCL constraints becomes the prerequisite for the formal and unambiguous development of the system [13].

1) Characteristics of OCL: Following are some of these:

a) *Both query and constraint language***:-**in UML version 1.1 OCL was used to specify constraints on the elements of the system. Constraint is a restriction on one or more value of the element of the software system. In UML version 2 OCL it is used to add additional information also.

b) *Mathematical foundation, but no use of mathematical symbol***:-**OCL is based on the theory of mathematical set, but it does not use any mathematical symbols.

c) *Declarative Language***:-** in declarative language an expression tells what should be done but it does not tell how it must be done. Through this a modeler can take high level decisions because modeler does not have to go into deep detail of how it must be done [11].

2) *Where to use OCL:* OCL can be used for different purposes:

a) It is used as a query language
b) It is used to specify invariants on classes
c) It is used to specify type invariant for stereotypes
d) It is used to describe pre and post conditions on operations and methods
e) It is used to describe guards
f) It is used to specify targets for messages and actions
g) It is used to specify constraints on operations
h) It is used to specify derivation rules for attributes for any expression over a UML model.

3) Types of OCL constraints

a) *Invariants:* Invariants are constraints which specify that the expression which is associated is always true.OCL invariant is a UML invariant which is stereotyped as “invariant”. Invariant must be declared in the context of UML classifier. Information like invariables and constraints can be easily expressed using OCL [8]. The constraint, stock price is greater than 0 is written as:

Context Company
Inv: Self.stockPrice () > 0

Context and self: OCL expressions are written in the form of the context of a specific type.

Eg: - context person

Reserved word self is used to refer to the contextual instance .If the context is person then self refers to the instance of person.

b) *Pre-conditions***:** These are the constraints which are associated with an operation or other behavioral feature [7]. Preconditions are the constraints in which an assumption is taken that it is true before the operation is executed.

Eg:-precondition constraint which is associated to operation income of person is:-

Context Person::income (): Integer

Pre: self. Age>=18

c) *Post-conditions***:** These are the constraints which are associated with the operation or other behavioral feature [7].post-conditions are the constraints that are satisfied after the operation is executed.

Eg:-post-condition associated to operation income of person

Context Person: income (): Integer
Post: result<5000

d) *Body Expression***:** This constraint is used to indicate the result of a query operation.

Eg:-income of a person is the sum of the salaries of her job

Context Person::income (): Integer
Body: self.job.salary->sum ()

This paper is organized as follows: section 2 describes about the work which is already done in this field. Section 3 shows the results of implementation of OCL constraints in java. Section 4 discusses the conclusion.

II. RELATED WORK

.**Ali et al. [1]** presents the approach of formalizing the UML class diagram and OCL constraints in the highly successful automated theorem prover Isabebelle using one of its built-in logics, HOL. The aim to do this is to create a formalization which will be accessible to the average software engineer and will also be able to prove consistency and other useful properties.

Briand et al. [2] in their paper “A Controlled Experiment on the impact of the object constraint language in UML-based Maintenance’ shows the results of the experiment that investigates the impact of the OCL on the maintainability of the UML diagrams. Results show that significant benefits can be obtained by combining OCL with UML diagrams because OCL is used to make UML models accurate, consistent and more precise.

Duarte [4] represents the argument that OCL is not special, being easily comparable to object-z.

Giese and Heldal [5] in their paper "From Informal to Formal Specifications in UML" presents a way to informal requirements to more formal specifications. In some projects there is a need of the informal description of the system so that people can easily understand the requirements but in some cases there is a need to make the requirements formal.

Levchenko [8] it presents creation of OCL evaluation framework, which provides almost complete and usable syntactic evaluation for OCL 2.0 and can perform semantic evaluation of all OCL expressions.

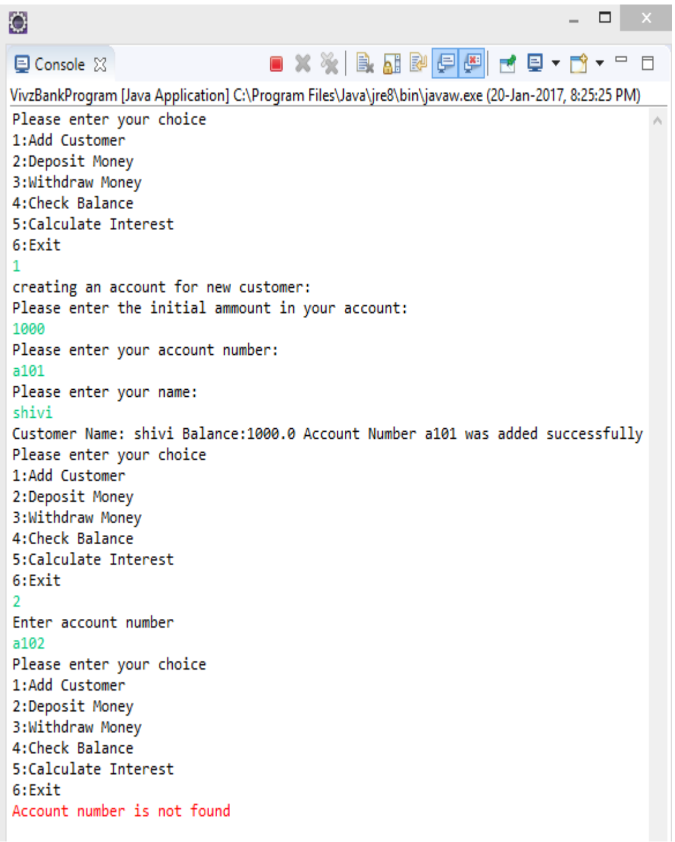
Moura et al. [9] presents the annotation of UML class diagram with fragments of object-z specification language. This approach is illustrated by modeling a simple object-oriented system as well as performing the required syntax checking and proof of desired properties.

Roe et al. [13] presents the mapping for translating systems modeled in the unified modeling language (UML) incorporating object constraint language (OCL) constraints into formal software specifications in object-z.

Tan et al. [14] in their paper "OCL Constraints Automatic Generation for UML Class Diagram" presents the way by which overall quality and efficiency of software design is enhanced by the use of UML as the standard modeling language for the software architecture design. But UML alone lacks in the formal specification so OCL is used along with UML.

III. Results of implementation of OCL constraints in java source code

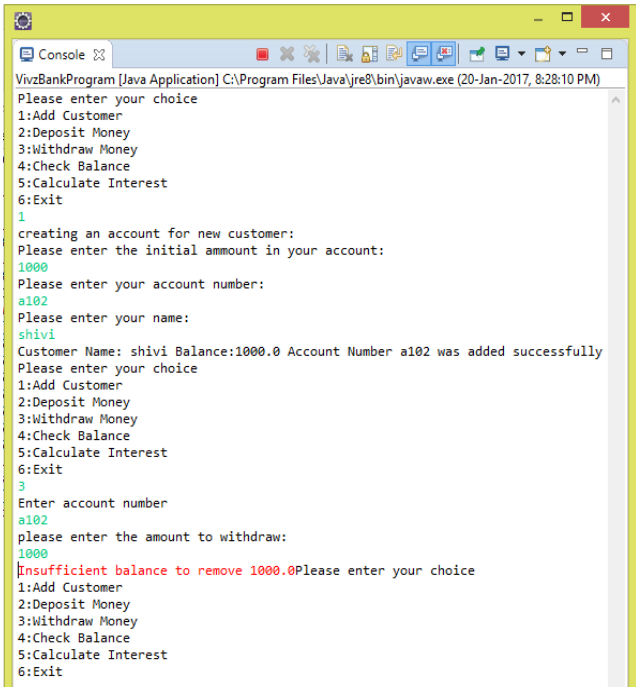
OCL constraints are implemented in java source code so as to map UML object oriented designs with OCL constraints to java classes and interfaces [7]. Constraints are defined in different components of the information system [15].This is example of results of OCL constraints written in source language. Above example is of bank account. A customer opens an account in the bank, the customer must have an initial balance of 100$ while opening the account. Customer can withdraw, deposit and check his balance at any time. The bank does not charge any fees for the first withdrawal but for all subsequent withdraws, bank charges some transaction fees. Bank also calculates a certain amount of interest on the amount deposited by the customer as per its interest rate.



```
VivzBankProgram [Java Application] C:\Program Files\Java\jre8\bin\javaw.exe (20-Jan-2017, 8:25:25 PM)
Please enter your choice
1:Add Customer
2:Deposit Money
3:Withdraw Money
4:Check Balance
5:Calculate Interest
6:Exit
1
creating an account for new customer:
Please enter the initial ammount in your account:
1000
Please enter your account number:
a101
Please enter your name:
shivi
Customer Name: shivi Balance:1000.0 Account Number a101 was added successfully
Please enter your choice
1:Add Customer
2:Deposit Money
3:Withdraw Money
4:Check Balance
5:Calculate Interest
6:Exit
2
Enter account number
a102
Please enter your choice
1:Add Customer
2:Deposit Money
3:Withdraw Money
4:Check Balance
5:Calculate Interest
6:Exit
Account number is not found
```

Figure 1: Precondition OCL Constraint

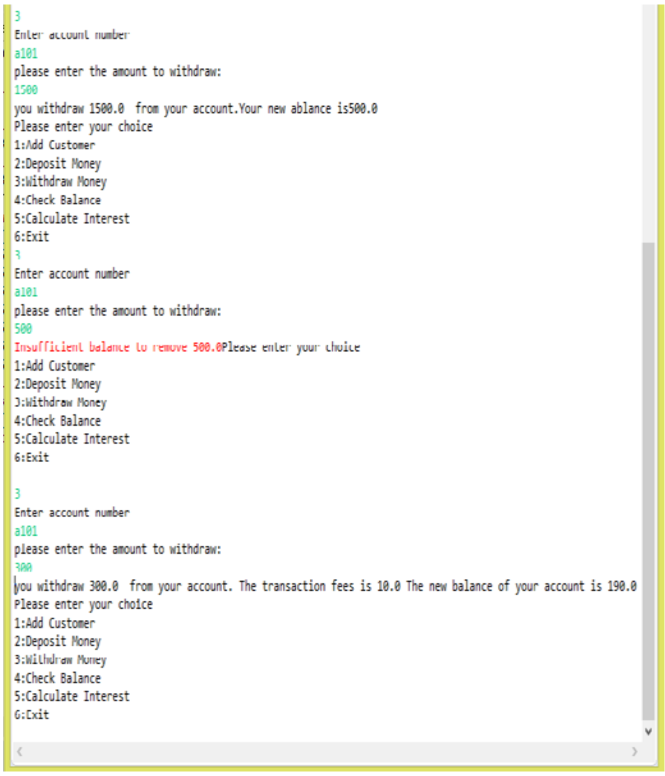
Figure 1 shows the result of applying the OCL constraint on bank class. It shows that if an account is not registered in the bank then it will not deposit any money and it will give an error message that Account number is not found.



```
VivzBankProgram [Java Application] C:\Program Files\Java\jre8\bin\javaw.exe (20-Jan-2017, 8:28:10 PM)
Please enter your choice
1:Add Customer
2:Deposit Money
3:Withdraw Money
4:Check Balance
5:Calculate Interest
6:Exit
1
creating an account for new customer:
Please enter the initial ammount in your account:
1000
Please enter your account number:
a102
Please enter your name:
shivi
Customer Name: shivi Balance:1000.0 Account Number a102 was added successfully
Please enter your choice
1:Add Customer
2:Deposit Money
3:Withdraw Money
4:Check Balance
5:Calculate Interest
6:Exit
3
Enter account number
a102
please enter the amount to withdraw:
1000
Insufficient balance to remove 1000.0Please enter your choice
1:Add Customer
2:Deposit Money
3:Withdraw Money
4:Check Balance
5:Calculate Interest
6:Exit
```

Figure 2: Constraint on Balance

Figure 2 shows the result of applying precondition constraint to the balance in the account. Precondition is that minimum balance in the account should be 100 rupees.



```
3
Enter account number
a101
please enter the amount to withdraw:
1500
you withdraw 1500.0  from your account.Your new ablance is500.0
Please enter your choice
1:Add Customer
2:Deposit Money
3:Withdraw Money
4:Check Balance
5:Calculate Interest
6:Exit
3
Enter account number
a101
please enter the amount to withdraw:
500
Insufficient balance to remove 500.0Please enter your choice
1:Add Customer
2:Deposit Money
3:Withdraw Money
4:Check Balance
5:Calculate Interest
6:Exit

3
Enter account number
a101
please enter the amount to withdraw:
300
you withdraw 300.0  from your account. The transaction fees is 10.0 The new balance of your account is 190.0
Please enter your choice
1:Add Customer
2:Deposit Money
3:Withdraw Money
4:Check Balance
5:Calculate Interest
6:Exit
```

Figure 3: Constraint on Transaction Fees

Figure 3 shows the constraint on transaction fees. On first transaction no transaction fees will be charged but on subsequent transactions transaction fees of 10 rupees will be charged.

1) OCL constraints used in the bank account example

a) *Pre-condition*: Customer must have an initial balance of 100$ which means that while withdrawing the money you have to ensure that no matter how much money the customer withdraws there is always 100$ that is remaining inside the account.
Eg:-private double balance=100;
Private double interestRate=8.5;
Private double transactionFees=10;
To deposit any money the number of customers of bank should be greater than 0

b) *Post-condition***:** Here post- condition is that Bank does not charge any fees for the first withdrawal but for all subsequent withdrawals, bank charges some transaction fees. So initially bank checks whether a customer is performing withdrawal for first time. if customer is performing withdrawal for first time then it will not issue any transaction fees but for all subsequent transactions it will charge a transaction fees. Post condition is that if the balance after withdrawal of money is less than 100 the message “insufficient funds to remove is displayed”

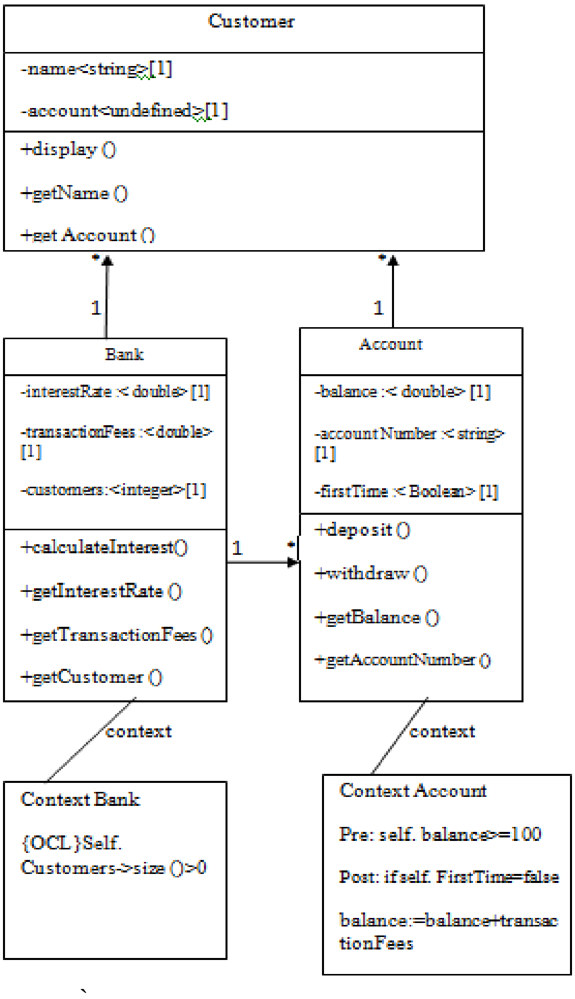


Figure 4: Class Diagram of Bank Account using OCL Constraints

IV. CONCLUSION

This paper presents the representation of OCL constraints in JAVA language. This representation is done for translating object-oriented models expressed in UML and OCL to JAVA classes and interfaces. This paper presents the importance of adding OCL constraints to the UML class Diagrams and how this addition of OCL constraints helps to accurately model the system. The Addition of these OCL constraints removes inconsistency and ambiguities of the `system and completely describes the system.

V. REFERENCES

[1] Ali, T., Nauman, M., & Alam, M. (2007). An Accessible Formal Specification of the UMLand OCL Meta-Model in Isabelle/HOL.

[2] Briand, L. C., Labiche, Y., & Yan, H. D. (2004). A Controlled Experiment on the Impact of the Object Constraint Language in UML-based Maintenance. Proceedings of the 20th IEEE International Conference on Software Maintenance. IEEE.

[3] Cabot, J., & Gogolla, M. (n.d.). Object Constraint Language (OCL):a Definitive Guide.

[4] Duarte, R., Junior, J., & Mota, A. (n.d.). Precise Modeling with UML: Why OCL?

[5] Giese, M., & Heldal, R. (n.d.). From Informal to Formal Specifications in UML.

[6] Hamie, A., Civcllo, F., Howse, J., Kent, S., & Mitchell, R. (1999). Reflections on the Object Constraint Language. 162-172.

[7] Hanada, K., Okano, K., Kusumoto, S., & Miyazawa, K. (n.d.). Practical Application of a Translation Tool from UML/OCL to Java Skeleton with JML Annotation.

[8] Levchenko, P. (2008). OCL Evaluation Framework:Integrated Environment for Syntactic and Semantic Evaluation of OCL Constraints.

[9] Moura, P., Borges, R., & Mota, A. (n.d.). Experimenting Formal Methods through UML.

[10] Nikiforova, O., Sejans, J., & Cernickins, A. (2011). Role of UML Class Diagram in Object-Oriented Software Development. 44.

[11] Normantas, K., & Vasilecas, O. (2009). MODELLING OF THE BUSINESS RULES USING UML/OCL. 1 (6), 2.1-2.10.

[12] Richters, M., & Gogolla, M. (2000). Validating UML Models and OCL Constraints. 265-277.

[13] Roe, D., Broda, K., & Russo, A. (n.d.). Mapping UML Models incorporating OCL Constraints into Object-Z.

[14] Tan, L., Yang, Z., & Xie, J. (2010). OCL Constraints Automatic Generation for UML Class Diagram.

[15] Weber, D., Szymanek, J., & Norrie, M. C. (n.d.). Unified OCL:Achieving System-Wide Constraint Representations.