


```

void Transaction(){

    string userInput;
    cin>>UserInput;

    if(UserInput == "Withdraw"){
        Withdraw();
    }else if(UserInput == "Deposit"){
        Deposit();
    }else{
        Balance();
    }
}

```

Fig. 8. The source code generated from EAD.

4. Discussion

MDA Tool such as EA⁷ (Enterprise Architecture) can generate a skeleton of a source code from a class diagram. In addition, EA can generate a source code from an activity diagram or a state machine diagram. However, The source code generated by EA does not have detail specification. It takes time and effort that the developer adds detail specification to the source code generated from the modified activity diagram.

The method can generate a source code including detail specification. The method can reduce time and effort to add detail specification to the source code generated from the modified activity diagram. Moreover, it can reduce time and effort to keep consistency between models and a source code after requirement specification is modified. Therefore, the method is useful for efficiency of software development.

5. Conclusion

This paper has proposed a modification method of a source code to correspond with a modified model in MDA. The method can generate the source code that corresponds with the modified activity diagram and has information about detail specification. We have confirmed that the method can generate a source code including the detail specification from the original activity diagram, the modified activity diagram, and the

original source code. Therefore, the method is useful for efficiency of software development.

Future issues are as follows.

- Development of the tool implemented the method
- Improvement of the method to treat with deletion or revision to source code and activity diagram.
- Improvement of the method to treat with other statements except if-statement.

References

1. MDA (Model Driven Architecture), <http://www.omg.org/mda> (accessed February 16, 2015).
2. Wada H, Yasutake Y, MDA (Model Driven Architecture) and Actual Development Process (in Japanese), UNISYS TECHNOLOGY REVIEW, No.61 (2004), pp. 47-59.
3. UML (Unified Modeling Language), <http://www.omg.org/spec/UML/2.4.1> (accessed February 15, 2015).
4. D. Lopes, S. Hammoudi, J. Bézivin, F. Jouault: Mapping Specification in MDA: From Theory to Practice, Interoperability of Enterprise Software and Applications, (Springer-Verlag London Ltd 2006), pp.253-264,
5. Ueno M, Omori M, An Approach to Keep Coherence between PIM and PSM of MDA (in Japanese), IPSJ SIG Technical Report, SE-156 (7), (2007), pp. 41-47.
6. A. Derezinska, Code Generation and Execution Framework for UML 2.0 Classes and State Machines, IMCSIT, (2008), pp. 517-524.
7. Enterprise Architect, <http://www.sparxsystems.jp> (accessed December 10, 2014).