

## Survey On Argo UML Tool

Tincy Rani<sup>1</sup>, Manisha Sanyal<sup>2</sup>, Sushil Garg<sup>3</sup>

CES Dept- RIMT-IET, CSE Dept- GGI-GIET, CSE Dept- RIMT-IET.

[Tisha11sep@yahoo.com](mailto:Tisha11sep@yahoo.com), [Manisha\\_Sanyal@yahoo.com](mailto:Manisha_Sanyal@yahoo.com), [sushilgarg70@yahoo.com](mailto:sushilgarg70@yahoo.com)

*Abstract— ArgoUML is an UML diagramming application written in Java and released under the open source Eclipse Public License. By virtue of being a Java application, it is available on any platform supported by Java. Within the open-source community, ArgoUML represents a major cooperative effort for the development of a UML development environment, ArgoUML is still under development, and processing errors arise periodically.*

**Index Terms—Argo, XMI, Use case, ATM machine.**

### I. INTRODUCTION

ArgoUML is a powerful yet easy-to-use interactive, graphical software design environment that supports the design, development and documentation of object-oriented software applications. If you are familiar with a family of software applications called Computer Aided Software Engineering (CASE) tools then you should find ArgoUML instantly familiar. The users of ArgoUML are software designers & architects, software developers, business analysts, systems Analysts and other professionals involved in the analysis, design and development of software applications.

### II. The Development of ArgoUML

During the 1980's a number of OOA&D process methodologies and notations were developed by different research teams [5]. It became clear there were many common themes and, during the 1990's, a unified approach for OOA&D notation was developed under the auspices of the Object Management Group. This standard became known as the Unified Modeling Language (UML), and is now the standard language for communicating OO concepts [6]. ArgoUML was conceived as a tool and environment for use in the analysis and design of object-oriented software systems. In this sense it is similar to many of the commercial CASE tools

that are sold as tools for modeling software systems. ArgoUML has a number of very important distinctions from many of these tools.

- It is free.
- ArgoUML draws on research in cognitive psychology to provide novel features that increase productivity by supporting the cognitive needs of object-oriented software designers and architects.
- ArgoUML supports open standards extensively - UML, XMI, SVG, OCL and others.
- ArgoUML is a 100% pure Java application. This allows ArgoUML to run on all platforms for which a reliable port of the Java platform is available.
- ArgoUML is an open source project. The availability of the source ensures that a new generation of software designers and researchers now has a proven framework from which they can drive the development and evolution of CASE tool technologies.

UML is the most prevalent OO modeling language and Java is one of the most productive OO development platforms [7]. Jason Robbins and the rest of his research team at the University of California, Irvine leveraged these benefits in creating ArgoUML. The result is a solid development tool and environment for OO systems design. Further, it provides a test bed for the evolution of object oriented CASE tools development and research. A first release of ArgoUML was available in 1998 and more than 100,000 downloads by mid-2001 show the impact that this project has made, being popular in educational and commercial fields.

### III. FEATURE OF ARGO TOOL

- Install with Java Web Start :- The ArgoUML website provides easy installation with Java Web Start.

- Runs on any platform with Java5 or Java6:- ArgoUML is written entirely in Java and uses the Java Foundation Classes. This allows ArgoUML to run on virtually any platform.

- Standard UML Meta model:- ArgoUML is compliant with the OMG Standard for UML 1.4. The core model repository is an implementation of the Java Metadata Interface (JMI) which directly supports MOF and uses the machine readable version of the UML 1.4 specification provided by the OMG.

- UML Profile support with profiles provided ArgoUML has support for creating profiles, and distributing models that reference common online available profiles. ArgoUML is delivered with profiles for: Java, C++, UML 1.4.

- UML Diagram Support

ArgoUML uses GEF, the UCI Graph Editing Framework to edit UML diagrams. The following diagram types are supported:

- ✓ Class diagram
- ✓ State chart diagram
- ✓ Activity diagram (including Swim lanes)
- ✓ Use Case diagram
- ✓ Collaboration diagram
- ✓ Deployment diagram (includes Object and Component diagram in one)
- ✓ Sequence diagram
- ✓ Adjustable font size.
- ✓ Diagram representation of stereotypes and datatypes
- ✓ Support for drawing Signals and Exceptions in class diagrams
- ✓ Configurable arrows for both-navigable associations.

- XMI Support:- XMI is an XML based exchange format between UML tools. ArgoUML uses this as standard saving mechanism so that easy interchange with other tools and compliance with open standards are secured. Additionally, exporting the model to XMI is possible. XMI version 1.0 was used for UML 1.3. ArgoUML 0.20 imports XMI 1.0 (by converting UML 1.3 to UML 1.4). And ArgoUML imports the UML1.4 formats XMI 1.1 & 1.2, but only writes XMI 1.2.

- Several diagram export formats :- Diagrams can be saved as GIF, PNG, PostScript, Encapsulated PS, PGML and SVG.

- Internationalization :- ArgoUML has been internationalized to American English, British English, French, German, Italian, Portuguese, Spanish, Russian, Norwegian Bokmål and Chinese.

- Diagram Editing

ArgoUML supports many diagram editing features that help you edit UML diagrams.

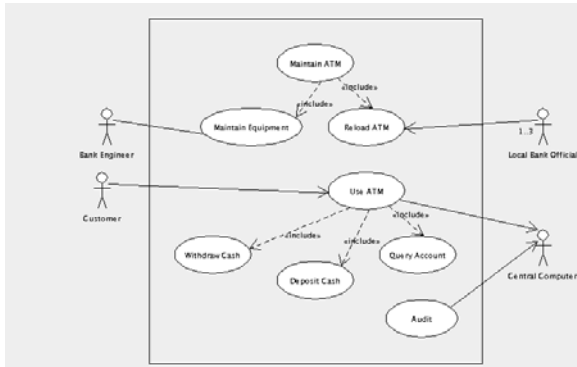
- Code Generation:- ArgoUML provides code generation for Java, C++, C#, PHP4 and PHP5. Other languages may be added since the code generation is a modular framework. The Java code generation works with the Java reverse engineering to provide basic round - trip engineering.

- Reverse Engineering:- ArgoUML provides a modular reverse engineering framework. Currently Java source code is provided by default and there are modules for Java Jar and class file import.

- Design Critics: - Design critics are simple agents that continuously execute in a background thread of control. They analyze the design as the designer is working and suggest possible improvements. These suggestions range from indications of syntax errors, to reminders to return to parts of the design that need finishing, to style guidelines, to the advice of expert designers. Many critics offer to automatically improve the design. Critics are controlled so that their suggestions are relevant and timely to the design task at hand, based on information in Argo's user model.

### IV. CASE STUDY:- ATM MACHIANE

In our ATM example we have identified “customer uses machine#”, “maintain machine# and “audit# as the three main use cases. We have identified “customer#, maintenance engineer“, #“local branch official# and “central computer# as the actors. The use cases are shown as ovals, the actors as stick people (even where they are machines), with lines (known as *associations* connecting use cases to the actors who are involved with them. A box around the use cases emphasizes the boundary between the system (defined by the use cases) and the actors who are external.



**Figure 4.1. Use case diagram for an ATM system**

In our ATM example so far we have just three use cases to describe all the behavior of the system. While use cases should always describe a significant chunk of system behavior, if they are too general they can be difficult to describe. We could for example define the behavior of the use case “Use ATM# in terms of the behavior of three simpler use cases, “Deposit Cash#, “Withdraw Cash# and “Query Account#. The main use case could be specified by *including* the behavior of the subsidiary use cases where needed. Similarly the “Maintain ATM# use case could be defined in terms of two use cases “Maintain Equipment# and “Reload ATM#. In this case the two actors involved in the main use case are really only involved in one or other of the two subsidiary use cases and this can be shown on the diagram. The decomposition of a use case into simpler sub-use cases is shown in UML by using an include *relationship*, a dotted arrow from the main use case to the subsidiary, with the label «include».

## V. Conclusion and Future Work

In this paper we have described some of the cognitive features of Argo/UML and key parts of the system’s design. These features are inspired by published theories of human cognition during design tasks. We have discussed the features in the context of Argo/UML, however, we believe them to be useful in many design contexts and tools. Our immediate research plans include the addition of new cognitive features to Argo/UML and evaluation of the impact of these features. In future work we develop a tool that generates the sequence of UML diagrams.

## REFERENCES

- [1] Al-Shamma, N., et al. Precision Graphics Markup Language(PGML): World Wide Web Consortium Note 10-April-1998. Available from <http://www.w3.org/TR/1998/NOTEPGML>.
- [2] Beck, K. and Johnson, R. Patterns generate architectures. Proc. European Conf. on Object-Oriented Programming(ECOOP’94). Bologna, Italy. 1994.
- [3] Bonnardel, N. and Sumner, T. Supporting evaluation in design: the impact of critiquing systems on designers of different skill levels. *Acta Psychologica*. vol. 91. 1996. pp.221-244.
- [4] Chun, H. W. and Lai, E.M.-K. Intelligent critic system for architectural design. *Trans. Knowledge and Data Engineering*. July 1997.
- [5] Fischer, G., Girgensohn, A., Nakakoji, K., Redmiles, D. Supporting Software Designers with Integrated, Domain-Oriented Design Environments, *IEEE Transaction on Software Engineering*, Special Issue: “Knowledge Representation and Reasoning in Software Engineering”, vol. 18, no. 6, pp. 511-522.
- [6] Gamma, E., Helm, R., Johnson, R., Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley. 1994.
- [7] Object constraint language specification. Object Management Group document ad/97-08-08. Sept. 1997. <http://www.omg.org/docs/ad/97-08-08.pdf>.
- [8] Riel, A. *Object-Oriented Design Heuristics*. Addison-Wesley: Reading, MA. 1996.
- [9] UML Revision Task Force. UML v1.3 R20 Meta model abstract syntax. Jan. 1999. Available from <http://uml.shl.com>
- [10] UML Semantics. Object Management Group document Ad/97-08-04. Sept. 1997. Available from <http://www.omg.org/docs/ad/97-08-04.pdf>.