

OpenOME: An Open-source Goal and Agent-Oriented Model Drawing and Analysis Tool

Jennifer Horkoff¹, Yijun Yu², Eric Yu³

¹Department of Computer Science, ³Faculty of Information, University of Toronto

²Department of Computing, Faculty of Maths, Computing & Technology,
The Open University

jenhork@cs.utoronto.ca, y.yu@open.ac.uk, eric.yu@utoronto.ca

Abstract. OpenOME is an Eclipse-based open-source tool supporting the construction and analysis of i* models. The tool is in a stable state and available freely for download. Recently added features include support for forward and backward interactive, qualitative i* analysis.

Keywords: Goal-and Agent-Oriented Models, Model Analysis, Interactive Analysis, Model Views, Tool Operability.

1 Introduction and Features

OpenOME is a tool for the creation and analysis of goal and agent-oriented models as part of a systems analysis process. The tool supports modeling of the social and intentional viewpoint of a system, allowing users to capture the motivations behind system development in a graphical form. Creation and analysis of agent-goal models supports requirements elicitation, exploration, communication and trade-off analysis. OpenOME is especially useful to support elicitation and analysis in early requirement phases, where important non-functional and social information is made explicit.

The tool allows users to draw models using the i* Framework syntax described in [1]. The tool allows users to create i* models graphically using a palette of shapes. Standard features such as saving, zoom, cut, copy, and paste are provided. Models are grouped under user-created projects, shown in a folder view. OpenOME imports and exports models in the GMF .ood and .oom format, as well the Q7 textual modeling language [2]. See Fig. 1 for a screenshot of the OpenOME interface. OpenOME supports the forward and backward interactive, qualitative i* analysis procedures described in [3,4]. Users can assign qualitative labels to intentions which represent their initial analysis question and then propagate these labels in a forward (direction of the link) or backward direction. The procedures are interactive, asking for user input to resolve conflicting or partial information. Results from multiple evaluations, including judgments made are stored, and can be viewed via the Alternatives Tab. Visualizations have been added to highlight model leaves and roots (potential starting points for analysis), areas of human judgment, and the intentions involved in a conflict in backward analysis.

Use of OpenOME, including the forward and backward evaluation procedures and current visualizations, has been tested in a series of user studies, reported in [5,6].

2 Development, Availability and Future Work

OpenOME development originated in OME, a desktop Java application developed at the University of Toronto to support i* modeling [7]. In 2004, development of OME ceased and the source code was ported to the Eclipse platform, creating an open source version taking advantage of the Eclipse Modeling and Graphical Modeling Frameworks (EMF & GMF). Use of these frameworks allows us to automatically generate model editing code from an i* metamodel. This code has been customized and expanded to support features specific to i* modeling. OpenOME architecture takes advantage of the Eclipse package development, allowing for extension or customization with the addition of a new development package. After several rounds of development, the current architecture no longer bears similarity to that of the OME tool. Current development is at version 3.4.1.

Windows, Linux and Mac releases of OpenOME can be downloaded from Sourceforge [8]. User documentation and tutorials are available under the User Links section here: <https://se.cs.toronto.edu/trac/ome/>. After several rounds of user studies, and through use of the tool in several systems analysis courses, OpenOME has reached a relatively mature and stable state. Bug reports and suggestions for new features can be sent to openome-support@cs.toronto.edu. More information can be found on the user web site [9] and on the developer wiki [10].

Development on general tool functionality and bug fixes is ongoing. We are currently working on expanding the tool in several directions, including import/export into iStarML [11], customizable syntax checking, tabular views of i* models, and features encouraging model iteration such as conflict checks amongst human judgments.

Acknowledgments. We thank many current and former graduate and undergraduate students for their work on the OpenOME tool.

References

- [1] E. Yu, "Towards modelling and reasoning support for early-phase requirements engineering," Proceedings of ISRE 97 3rd IEEE International Symposium on Requirements Engineering, vol. 97, 1997, pp. 226-235.
- [2] D.P. Leite, J.C. Sampaio, Y. Yu, L. Liu, E.S.K. Yu, and J. Mylopoulos, "Quality-Based Software Reuse," CAISE, vol. 3520, 2005, pp. 535-550.
- [3] J. Horkoff and E. Yu, "Finding Solutions in Goal Models: An Interactive Backward Reasoning Approach," Proc. 29th Int. Conference on Conceptual Modeling (ER 2010) Vancouver BC Canada November 2010, Springer-Verlag, 2010, p. 59.
- [4] J. Horkoff and E. Yu, "A Qualitative, Interactive Evaluation Procedure for Goal- and Agent-Oriented Models," CAiSE'09 Forum, Vol-453, CEUR-WS.org, 2009, pp. 19-24.
- [5] J. Horkoff and E. Yu, "Interactive Goal Model Analysis Applied - Systematic Procedures versus Ad hoc Analysis," The Practice of Enterprise Modeling, 3rd IFIP WG8.1 (PoEM'10), 2010.
- [6] J. Horkoff and E. Yu, "Visualizations to support interactive goal model analysis," Requirements Engineering Visualization REV 2010 Fifth International Workshop on, IEEE, 2010, p. 1-10.
- [7] "OME, Organization Modelling Environment," 2008.
- [8] "Sourceforge: OpenOME," <http://sourceforge.net/projects/openome>, 2011.
- [9] "OpenOME, an open-source requirements engineering tool," <http://www.cs.toronto.edu/km/openome/>, 2010.
- [10] "OpenOME Trac Wiki," <https://se.cs.toronto.edu/trac/ome/>, 2011.
- [11] "iStarML," <http://www.upc.edu/gessi/istarmil/>, 2011.