VI. CONCLUSION AND FUTURE WORK

An approach is proposed to generate test cases automatically from UML sequence diagram, class diagram and OCL. The approach generates efficient test cases that meet required coverage criteria. Furthermore, the relationship between edges has been recognised by the edge relationships table. This table is generated automatically to reveal the relationship of the edges as specified in the sequence diagram. The proposed methodology covers *loop*, *parallel*, *alternative*, *option*, *break* and *sequence* relationships and a number of structural specifications as well. This work provides an efficient technique to generate testing scenarios graph that is used to represent the testing scenarios of the events in a sequence diagram. The testing scenario graph represents combined fragments and shows the relationships among scenarios. The proposed methodology decomposes testing scenarios graph to combined test scenarios to achieve minimum number of testing scenarios. Information Table (**I**nf**T**) technique is proposed to provide the sequence diagram with exact method signatures that are obtained from the class diagram. Despite this, presence of the redundant edges in each testing scenario led us to develop a new technique to select best testing scenario. To achieve this, we have used *All messages coverage criterion* and *Combined Fragments coverage criterion* to evaluate the best testing scenario (and the best test case) using GA. The weakness that appeared in selecting best testing scenario is a variety of the coverage percentage that the best testing scenario can cover. This variety comes from the variety of the model designs. Generally, this technique almost covers more than 85% of Combined Fragment's edges and more than 94% of the sequence diagram messages.

The test cases that are generated automatically in this paper meet the coverage criteria. However, UML sequence diagrams do not contain all information related to verification and non-functional parameters of the software. This limitation comes from the semi-formal characteristics of UML diagrams [12], especially during the first cycle of the software production. In order to avoid its semi-formal characteristics, we propose to transform sequence diagram and class diagram to HLPN, a type of Petri Net in future. Petri Net (PN) is a graphical diagram for the formal description of the flow of activities in complex systems [13].

REFERENCES

[1] A. Alhroob, K. Dahal, and H. Alamgir, “Automatic Test Cases Generation from Software Specifications Modules," in Proceedings of the 4th IFIP TC2 Central and East European. Conference on Software Engineering Techniques CEE-SET. Springer, 2009, pp. 130-142.

[2] A. Cavarra and J. Kuster-Filipe, “Combining sequence diagrams and OCL for liveness," Electronic Notes in Theoretical Computer Science, vol. 115, pp. 19-38, 2005.

[3] A. O utt, Z. Jin, and J. Pan, “The dynamic domain reduction procedure for test data generation," Software: Practice and Experience, vol. 29, no. 2, pp. 167{193, 1999.

[4] [4]B. Li, Z. Li, L. Qing, and Y. Chen, “Test Case Automate Generation from UML Sequence Diagram and OCL Expression," in Proceedings of the 2007 International Conference on Computational Intelligence and Security. IEEE Computer Society, 2007, pp. 1048-1052.

[5] G. Booch, J. Rumbaugh, and I. Jacobson, UniedModeling Language User Guide, The (Addison-Wesley Object Technology Series), 2005.

[6] D. Aredo, “A framework for semantics of UML sequence diagrams in PVS," Journal of Universal Computer Science, vol. 8, no. 7, pp. 674{697, 2002.

[7] X. Li, Z. Liu, and H. Jifeng, “A Formal Semantics of UML Sequence Diagram," in Proceedings of the 2004 Australian Software Engineering Conference. IEEE Computer Society, 2004, p. 168.

[8] OMG, “Object constraintlanguage, version 2.2," http://www.omg.org/spec/OCL/2.2.htm [Last accessed in August 10, 2010],2010. [Online]. Available: http://www.omg.org/spec/OCL/2.2.htm

[9] N. Koochakzadeh and V. Garousi, “A Tester-Assisted Methodology for Test Re-dundancy Detection," Journal on Information and Software Technology, vol. 52, no. 5, pp. 625-640, 2010.

[10] J. Edvardsson, “A survey on automatic test data generation," in Proceedings of the 2nd Conference on Computer Science and Engineering, 1999, pp. 21{28.

[11] A. Andrews, R. France, S. Ghosh, and G. Craig, “Test adequacy criteria for UML design models," Software Testing Verication and Reliability, vol. 13, no. 2, pp. 95-127, 2003.

[12] H. Motameni, A. Movaghar, I. Daneshfar, and H. Zadeh, “Mapping to Convert Activity Diagram in Fuzzy UML to Fuzzy Petri Net," World Applied Sciences Journal, vol. 3, no. 3, pp. 514-521, 2008.

[13] Bobbio, “System modelling with Petri nets," in Systems reliability assessment: proceedings of the Ispra course held at the EscuelaTecnica Superior de IngenierosNavales, Madrid, Spain, September 19-23, 1988, in collaboration with Universidad Politecnica de Madrid. Springer, 1990, p. 103.

[14] DebasishKundu, DebasisSamanta, Rajib Mall "Automatic code generation from unified modelling language sequence diagrams", IET Software, Volume 7, Issue 1, February 2013, p. 12 – 28.

[15] Federico Ciccozzi, Antonio Cicchetti, Toni Siljamäki, "Automating Test Cases Generation: From xtUML System Models to QML Test Models", ACM proceeding in MOMPES ’10, September 20, 2010, Antwerp, Belgium

[16] PreetiGulia, R. S. Chillar, "A New Approach to Generate and Optimize Test Cases for UML State Diagram". ACM SIGSOFT Software Engineering Notes archive, Volume 37 Issue 3, May 2012, Pages 1-5

[17] Mahesh Shirole, AmitSuthar, Rajeev Kumar, Generation of Improved Test Cases from UML State Diagram Using Genetic Algorithm, Proceeding of the 4th India Software Engineering Conference, ISEC 2011, Pages 125-134

[18] ValentinDallmeier, Nikolai Knopp, Christoph Mallon, Gordon Fraser, Sebastian Hack, and Andreas Zeller, Member, "Automatically Generating Test Cases for Specification Mining", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 38, NO. 2, MARCH/APRIL 2012

[19] TarkanTekcan, Vladimir Zlokolica, VukotaPekovic, Nikola Teslic and Mustafa Gündüzalp, "User-driven Automatic Test-case Generation for DTV/STB Reliable Functional Verification", IEEE Transactions on Consumer Electronics, Vol. 58, No. 2, May 2012. Pages: 587-595

[20] Gordon Fraser and Andreas Zeller, "Mutation-Driven Generation of Unit Tests and Oracles", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 38, NO. 2, MARCH/APRIL 2012, Pages: 287-292

[21] Shaukat Ali, Lionel C. Briand, HadiHemmati, and Rajwinder K. Panesar-Walawege "A Systematic Review of the Application and Empirical Investigation of Search-Based Test Case Generation", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 36, NO. 6, NOVEMBER/DECEMBER 2010