plications and is based on dynamic analysis of web application execution. Neither of these tools enable users to target specific client-side features, nor do they filter the generated scenarios in order to remove the unnecessary ones.

There are also tools that support the understanding of dynamic web page behavior: FireCrystal [13] and Script InSight [7]. FireCrystal supports the understanding of interactive behaviors by recording user interactions and logging DOM changes, user generated events, and JavaScript code executions. The user can then use a time-line to study the code executed for the particular behavior. Script InSight relates the elements in the page with the lower-level JavaScript syntax, by gathering data during the script's execution and building a context-sensitive, control-flow model with tracing information. Compared to our approach they make no attempts to track data dependencies between different code expressions, nor to identify individual features in the analyzed code.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a prototype open-source tool for client-side web application analysis and reuse – Firecrow. The tool provides the functionality of automatic usage scenario generation, feature identification, and feature integration. These three functionalities target the general goal of facilitating reuse in the domain of client-side web applications. The tool is developed as a JavaScript library and can be used from different browsers: Firefox, Chrome, Safari, PhantomJs. However, due to the advantages of the Firefox platform, the tool is integrated into the Firefox browser, where it can be used as a plugin to Firefox Web Developer Tools.

For future work, we plan to explore the usage of some of its functionalities for different purposes than reuse, e.g. feature identification could be used to facilitate debugging, code understanding, maintenance, and even for deriving various software metrics. Web applications in general are composed out of two parts: the server-side application which implements business logic and data access, and the client-side which represents the UI of the application. In its current state, Firecrow deals exclusively with the client-side, and as part of future work, we plan to extend it in order to support server-side applications also.

5. REFERENCES

[1] H. Agrawal and J. R. Horgan. Dynamic program slicing. *ACM SIGPLAN Notices*, 25(6):246–256, 1990.

[2] S. Artzi, J. Dolby, S. H. Jensen, A. Møller, and F. Tip. A framework for automated testing of Javascript web applications. In *Software Engineering, ICSE 2011, 33rd International Conference on*, pages 571–580. ACM, 2011.

[3] V. R. Basili, L. C. Briand, and W. L. Melo. How reuse influences productivity in object-oriented systems. *Communications of the ACM*, 39(10):104–116, 1996.

[4] W. B. Frakes and G. Succi. An industrial study of reuse, quality, and productivity. *Journal of Systems and Software*, 57(2):99–106, 2001.

[5] R. Holmes and R. J. Walker. Semi-Automating Pragmatic Reuse Tasks. Automated Software Engineering, pages 481–482. IEEE Computer Society, 2008.

[6] C. W. Krueger. Software reuse. *ACM Computing Surveys (CSUR)*, 24(2):131–183, 1992.

[7] P. Li and E. Wohlstadter. Script insight: Using models to explore Javascript code from the browser view. In *International Conference on Web Engineering*, pages 260–274, 2009.

[8] J. Maras. *Automating Reuse in Web Application Development*. PhD thesis, Mälardalen University, April 2014.

[9] J. Maras, J. Carlson, and I. Crnković. Towards automatic client-side feature reuse. In *Web Information Systems Engineering–WISE 2013*, pages 479–488. Springer, 2013.

[10] J. Maras, M. Štula, and J. Carlson. Generating feature usage scenarios in client-side web applications. In *Web Engineering*, pages 186–200. Springer, 2013.

[11] J. Maras, M. Stula, J. Carlson, and I. Crnkovic. Identifying code of individual features in client-side web applications. 2013.

[12] M. D. McIlroy, J. Buxton, P. Naur, and B. Randell. Mass-produced software components. In *Proceedings of the 1st International Conference on Software Engineering, Garmisch Pattenkirchen, Germany*, pages 88–98. sn, 1968.

[13] S. Oney and B. Myers. FireCrystal: Understanding interactive behaviors in dynamic web pages. In *IEEE Symposium on Visual Languages and Human-Centric Computing*, pages 105–108. IEEE Computer Society, 2009.

[14] D. L. Parnas. On the design and development of program families. *Software Engineering, IEEE Transactions on*, (1):1–9, 1976.

[15] P. Saxena, D. Akhawe, S. Hanna, F. Mao, S. McCamant, and D. Song. A symbolic execution framework for Javascript. In *Security and Privacy (SP), 2010 IEEE Symposium on*, pages 513–528. IEEE, 2010.

[16] M. Schraefel, Y. Zhu, D. Modjeska, D. Wigdor, and S. Zhao. Hunter Gatherer: Interaction Support for the Creation and Management of Within-Web-Page Collections. World Wide Web, pages 172–181, 2002.

[17] K. Sen, D. Marinov, and G. Agha. *CUTE: a concolic unit testing engine for C*, volume 30. ACM, 2005.

[18] T. A. Standish. An essay on software reuse. *Software Engineering, IEEE Transactions on*, (5):494–497, 1984.

[19] G. Succi, L. Benedicenti, and T. Vernazza. Analysis of the effects of software reuse on customer satisfaction in an RPG environment. *Software Engineering, IEEE Transactions on*, 27(5):473–479, 2001.

[20] A. Sugiura and Y. Koseki. Internet scrapbook: creating personalized world wide web pages. Human Computer Interaction, pages 343–344. ACM, 1997.

[21] Y. Tanaka, K. Ito, and J. Fujima. Meme Media for Clipping and Combining Web Resources. *World Wide Web*, 9:117–142, 2006.