

Modellierung von Softwaresystemen und Programmiersprachen

Ort: Kettenhofweg 139, EG, Seminarraum 5 **Virtuell**

Vortrag: ca. 30 min Vortrag + ca. 10 min Diskussion

Ablauf:

Die SeminarteilnehmerInnen erhalten eine Originalveröffentlichung (PDF-Datei) und ggf. Software, mit der sich auseinandergesetzt werden soll. Das Thema der Veröffentlichung soll als Vortrag präsentiert werden.

Themen und Artikel:

1. A) Selic, B. (2003). The pragmatics of model-driven development. *IEEE Software*, 20(5), 19–25. <https://doi.org/10.1109/MS.2003.1231146>
B) Atkinson, C., & Kühne, T. (2003). Model-driven development: A metamodeling foundation. *IEEE Software*, 20(5), 36–41. <https://doi.org/10.1109/MS.2003.1231149>
2. France, R., & Rumpe, B. (2007). Model-driven development of complex software: A research roadmap. *FoSE 2007: Future of Software Engineering*, (May 2007), 37–54. <https://doi.org/10.1109/FOSE.2007.14>
3. Kats, L. C. L., & Visser, E. (2011). The spoofax language workbench. *ACM SIGPLAN Notices*, 45(10), 444. <https://doi.org/10.1145/1932682.1869497>
4. De Jonge, M., & Visser, E. (n.d.). An Algorithm for Layout Preservation in Refactoring Transformations. Retrieved from <http://www.se.ewi.tudelft.nl/techreports/>
5. Paige, R. F., Kolovos, D. S., & Polack, F. A. C. (2014). A tutorial on metamodelling for grammar researchers. In *Science of Computer Programming* (Vol. 96, pp. 396–416). Elsevier. <https://doi.org/10.1016/j.scico.2014.05.007>
6. Hopper, G. M. (1952). The education of a computer. In *Proceedings of the 1952 ACM national meeting (Pittsburgh) on - ACM '52* (pp. 243–249). New York, USA: ACM Press. <https://doi.org/10.1145/609784.609818>
7. Allen, F. E. (1970). Control Flow Analysis. *Proceedings of ACM Symposium on Compiler Optimization*, 1–19. Retrieved from <http://portal.acm.org/citation.cfm?doid=800028.808479>
8. Parr, T., Harwell, S., & Fisher, K. (2014). Adaptive LL(*) parsing. *ACM SIGPLAN Notices*, 49(10), 579–598. <https://doi.org/10.1145/2714064.2660202>
9. Hoffer, M., Poliwoda, C., & Wittum, G. (2013). Visual reflection library: a framework for declarative GUI programming on the Java platform. *Computing and Visualization in Science*, 16(4). <https://doi.org/10.1007/s00791-014-0230-y>

10. Foster, J. N., Greenwald, M. B., Moore, J. T., Pierce, B. C., & Schmitt, A. (2007). Combinators for bidirectional tree transformations. *ACM Transactions on Programming Languages and Systems*, 29(3), 17-es. <https://doi.org/10.1145/1232420.1232424>
11. Sutherland, W. R. (1966). The on-line graphical specification of computer procedures. Massachusetts Institute of Technology.

Ansprechpartner für diese Themen:

Dr. Michael Hoffer **E-Mail:** info@michaelhoffer.de