Reflections on the Birth of Spoofax

Karl Trygve Kalleberg ✉ 🏠 🆔

KolibriFX AS, Oslo, Norway

─── **Abstract** ───────────────────────────

The Spoofax Language Workbench is one of the many successful research projects to come out of Eelco's tremendously productive career, as evidenced by the Most Influential Paper designation awarded at OOPSLA 2020 to the famous 2010 paper by Lennart and Eelco [6]. In connection with the award, Eelco wrote an excellent retrospective [9].

Here, I will reflect briefly on the birth of the Spoofax, subject to omissions and inaccuracies, as the topic may be of interest to hardcore fans. Eelco was my PhD co-supervisor, and I was fortunate to be part of his group during the inception of the project.

A little-known fact is that Spoofax did not actually start out as a language workbench. Its humble beginnings were as an interactive development environment (IDE) for Stratego/XT [1]. During my time with Eelco in Utrecht, he and I had numerous walks in the forest around De Uithof, plotting the total world domination of program transformation in general and Stratego/XT in particular. Whenever interactive transformations came up (refactoring, code completion, semantics-aware navigation, live compilation), I always remarked that we could not really support these use cases with the batch-based Stratego/XT system we had at the time. As some of you know, Eelco was a lover of music, but not into broken records, so I was politely encouraged to put up or shut up.

At the time, building an entire platform for interactive program transformation seemed daunting to me, so I settled on the easier goal of making an interactive editing environment for Stratego based on Eclipse. As with any new project, choosing the name was the hardest part, but once I came up with a pronounceable word that was also free domain name, I registered `spoofax.{com,net,org}` in February of 2005, and then I was off to the races. The first version (0.1.0) of Spoofax was released on March 12 2005, and featured rudimentary syntax highlighting. Over the course of the next few months, I added features such as simple code completion of rule and strategy names, outline, and parenthesis matching (Figures 1a and 1b).

Looking for peer review, Eelco suggested we drive over to Günter Kniesel in Bonn and present what we had at a workshop there, so we did. That was the first time we showed Spoofax to the world. At the workshop, the concept of a language workbench came up again – only we did not use that term yet – but we were very far from realizing anything like that at the time. After summer, Eelco and I had a long discussion on where to go from here. We decided it made sense to reimplement the necessary parts of Stratego/XT in Java, to enable tighter, in-process, integration of the Stratego/XT universe with the Eclipse IDE ecosystem.
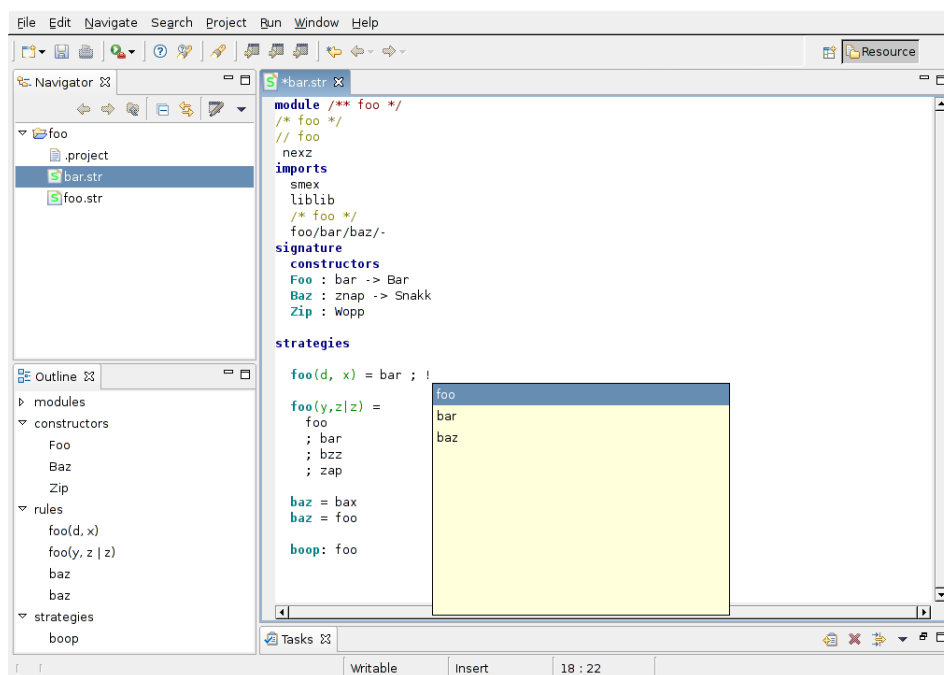
(a) Spoofax version 0.2.0, April 2005.
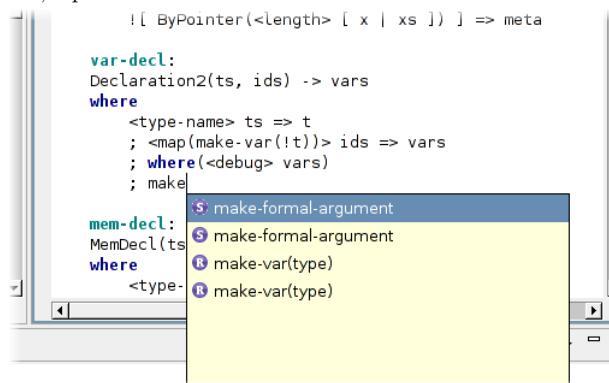
(b) Spoofax version 0.3.0, May 2005.

Figure 1 Screenshots from the early days of Spoofax.

I first put together a Stratego interpreter and then an implementation of SGLR in Java in the autumn of 2005. I plugged both into the rudimentary Spoofax implementation I had already built. This was the first time we had something that could conceivably become the basis of a language workbench, albeit in the crudest form possible. There were a number of really bad kinks. The source code had to be syntactically correct for the parser to work, because, as we all know, parsing stinks. Extending the platform with new transformation logic required manually executing the Stratego compiler on the command line with some unusual switches, then placing the output files in a particular directory monitored by the IDE. An unwieldy tool, with a lot of sharp edges to cut yourself. But it was a start.

My original motivation was for Spoofax to be a modern IDE for Stratego/XT, one that you could extend with your own transformations written in Stratego. In the beginning, Eelco was not convinced that this was a good idea. I cannot know for sure exactly why that was

the case, but I remember him as very happy with Emacs and I know that he had limited experience with modern IDEs at the time. Moreover, his previous involvement with the ASF+SDF meta-environment [8] had taught him how difficult and expensive it is to build and maintain desktop applications with graphical user-interfaces. Over time, I believe he saw that, with Eclipse, the UI part came "mostly for free". After we (Lennart, mostly) put together the `sdf2imp` [5] prototype which made it possible to plug any language with an SDF grammar into Eclipse, I believe Eelco realized that he could focus on all of the fun parts from the ASF+SDF meta-environment, with hardly any of the dreaded UI maintenance. This enabled him to formulate a much bigger vision for what this could become, and he leaned on that vision as he guided the project forward after my departure from the group.

Across the gulf of time, minds immeasurably superior to mine regarded the Spoofax prototype with curious eyes, and slowly and surely, they drew their plans to improve it [10]. To name some and omit many (sorry): Lennart single-handedly evolved the early proof-of-concept into a real platform – what would become an actual language workbench [4]; Maartje extended the SGLR implementation and with support for parsing incomplete syntax [2], finally making it practical inside an editor; Gabriel took the concept even further by developing language-parametric methods for deriving interactive programming systems automatically from high-level specifications [7].

It is a testament to his foresight and perseverance that Eelco was able to steadily grow the project from a somewhat programmable Stratego editor into the extensive language ecosystem that Spoofax is today. His knack for attracting talented and hard-working students, and letting them loose on some of the gnarliest problems in the field has been a good recipe for progress. One of the traits I most appreciated in Eelco was his keen balance between the mindset of an engineer and an academic. Constantly the applied computer scientist: pragmatic, practical and visionary. I always found it inspiring to be around him.

After my PhD [3], life took me on a different path, but Eelco and I kept in touch from time to time. I was last in touch with him in late 2021, when we agreed that I would transfer the ownership of the original Spoofax internet domains to his custody. This had been on the docket ever since I left in 2007, but we never got around to it before.

Going forward, it is my hope that the community around Spoofax will continue to thrive in the years ahead, like an epic sax tune that goes on forever. ♪*Never gonna give you up* ♪.

So long, and thanks for all the transformations. I will miss you, my friend.

References

1　Martin Bravenboer, Karl Trygve Kalleberg, Rob Vermaas, and Eelco Visser. Stratego/XT 0.17. A language and toolset for program transformation. *Science of Computer Programming*, 72(1-2):52–70, 2008. `doi:10.1016/j.scico.2007.11.003`.

2　Maartje de Jonge, Lennart C. L. Kats, Eelco Visser, and Emma Söderberg. Natural and flexible error recovery for generated modular language environments. *ACM Transactions on Programming Languages and Systems*, 34(4):15, 2012. `doi:10.1145/2400676.2400678`.

3　Karl Trygve Kalleberg. *Abstractions for Language-Independent Program Transformations*. PhD thesis, University of Bergen, 2007.

4　Lennart C. L. Kats. *Building Blocks for Language Workbenches*. PhD thesis, Delft University of Technology, 2011.

5　Lennart C. L. Kats, Karl Trygve Kalleberg, and Eelco Visser. Generating editors for embedded languages. integrating sglr into imp. In Jurgen J. Vinju and Adrian Johnstone, editors, *Proceedings of the Eight Workshop on Language Descriptions, Tools, and Applications*, volume 238(5) of *Electronic Notes in Theoretical Computer Science*. Elsevier, April 2008.

6   Lennart C. L. Kats and Eelco Visser. The Spoofax language workbench. In William R. Cook, Siobhán Clarke, and Martin C. Rinard, editors, *Companion to the 25th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, SPLASH/OOPSLA 2010, October 17-21, 2010, Reno/Tahoe, Nevada, USA*, pages 237–238. ACM, 2010. `doi:10.1145/1869542.1869592`.

7   Gabriël Konat. *Language-Parametric Methods for Developing Interactive Programming Systems*. PhD thesis, Delft University, 2019.

8   Mark G. J. van den Brand, Arie van Deursen, Jan Heering, H. A. de Jong, Merijn de Jonge, Tobias Kuipers, Paul Klint, Leon Moonen, Pieter A. Olivier, Jeroen Scheerder, Jurgen J. Vinju, Eelco Visser, and Joost Visser. The ASF+SDF meta-environment: A component-based language development environment. In Reinhard Wilhelm, editor, *Compiler Construction, 10th International Conference, CC 2001 Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2001 Genova, Italy, April 2-6, 2001, Proceedings*, volume 2027 of *Lecture Notes in Computer Science*, pages 365–370. Springer, 2001. `doi:10.1016/S1571-0661(04)80917-4`.

9   Eelco Visser. A Brief History of the Spoofax Language Workbench. `https://eelcovisser.org/blog/2021/02/08/spoofax-mip/`, 2021. [Online; accessed 30-January-2023].

10   Jeff Wayne. The Eve of the War (track 1), The Musical Version of The War of the Worlds. CBS, 1978. Produced by Jeff Wayne.