

Eelco Visser – An Exceptional SLE Researcher

Ralf Lämmel   

Universität Koblenz, Germany

Abstract

These notes honor Eelco Visser as an exceptional member of the *Software Language Engineering* (SLE) community. The notes are authored from the perspective of an SLE co-founder and a continuous SLE supporter. As an inevitable side effect, a short history of SLE is captured. The commemoration begins with Eelco’s role in launching the SLE conference. The commemoration continues with Eelco’s contributions as an author and his involvement with running the conference and working towards an *SLE Body of Knowledge* (SLEBoK). The commemoration ends with recalling Eelco’s role as the de-facto SLE photographer.

2012 ACM Subject Classification Software and its engineering → Software notations and tools

Keywords and phrases Eelco Visser, SLE, SLE conference, Software Language Engineering, SLEBoK

Digital Object Identifier 10.4230/OASICS.EVCS.2023.17

1 Eelco Visser’s role in launching the SLE conference

The *Software Language Engineering* (SLE) conference^{1,2} had its first edition in 2008, but it was getting off the ground in 2007 by an announcement at the ATEM 2007 workshop [13]³ at MODELS 2007. Formally, SLE was founded by a joint effort of the ATEM and LDTA communities.^{4,5}

The ATEM organizers were very well aware of Eelco Visser’s potential role in making the SLE launch successful. This was one of the reasons why he was invited as a panelist; see Fig. 1 for the final email that was sent to the ATEM 2007 panelists.

In 2007, the broader PL/SE/MODELS+ community – to the extent it was even aware of the emerging SLE effort – was not sure about the scope or the unique role of SLE in the landscape of venues and communities. To many, it felt that there were just enough conferences already to cover all SLE topics. Obviously, the SLE founders felt differently.

These birth pains are also somewhat visible from the questions for the panelists, as we tried to gather additional expert support for the SLE cause. Eelco prepared notes for the panel; see Fig. 2. (The notes are formatted and elisions are applied to facilitate the inclusion into the paper at hand.) On the grounds of the figure, the following claims can be supported:

1. Eelco’s definition of SLE (Q1) demarcates SLE from POPL and PLDI in that SLE is understood as not focusing on foundations and implementation of programming languages. I assume that the mentioning of “abstractions” hints at DSLs (see also Eelco’s answer to

¹ <https://www.sleconf.org>

² <https://dblp.org/db/conf/sle/index.html>

³ ATEM is META (the subject of MODELS) backwards (in “reverse”); ATEM started off with reverse engineering focus on metamodels and schemas to arrive (by 2006) at the much broader scope of SLE with coverage of modelware, schemaware, grammarware, and ontoware.

⁴ More precisely, based on the composition of the initial SLE steering committee, SLE was founded by a combination of the ATEM representatives (Jean-Marie Favre, Dragan Gasevic, Ralf Lämmel, and Andreas Winter) and LDTA representatives (Mark van den Brand, Görel Hedin, and Eric Van Wyk) as well as James Cordy. ATEM was immediately “subsumed” by SLE in 2008, whereas LDTA’s subsumption by SLE was also eventually completed – after a few more LDTA workshop editions.

⁵ <http://www.sleconf.org/2008/>



© Ralf Lämmel;
licensed under Creative Commons License CC-BY 4.0

Eelco Visser Commemorative Symposium (EVCS 2023).

Editors: Ralf Lämmel, Peter D. Mosses, and Friedrich Steimann; Article No. 17; pp. 17:1–17:13

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

From: R. Lämmel <...>
 Date: Wed, Sep 26, 2007 at 12:52 AM
 Subject: Input on the ATEM panel
 To: E. Visser <...>, T. Kuehne <...>, A. Kleppe <...>, K. Czarnecki <...>
 Cc: A. Winter <...>, D. Gasevic <...>, J.-M. Favre <...>

Dear panelists,

we look forward the panel on “Grand Challenges in Software Language Engineering”, which is going to be the closing event at ATEM 2007, which is the 4th International Workshop on (Software) Language Engineering (SLE). The following text is meant to help you preparing for the panel.

[Parts elided – RL]

I will moderate the panel.
 We have collected some questions that you may find inspiring.
<https://professor-fish.blogspot.com/2022/11/atem-2007-panel-post-rehosted.html>
 [Original link replaced – RL]

Thanks and regards,
 Ralf

■ **Figure 1** Invitation of Eelco Visser to the ATEM 2007 panel.

- Q6) and the mentioning of “programmer productivity” hints at tools other than common PLDI artifact types. Arguably, Eelco’s definition of SLE (Q1) does not directly get into an explanation of the “engineering” bit, but see Eelco’s answer to the next question (Q2).
2. Eelco was involved in DSL-related research since his PhD times. Ever since the early work on Stratego [34], Eelco was getting interested in translation, generation, and optimization for DSLs. His belief in the importance of this area shows in the answer to the question what a panelist would regard as a current SLE challenge (Q3). Eelco nominates “safety of code generation” and he points out that GPCE has knowledge in this area, but industry does not use it, which may suggest that “engineering” has to be addressed in this context. The MDA-related responses (Q5 and Q6) also engage with code generation.
 3. When being asked about linguistics and usability (Q7), Eelco is more interested in usability than linguistics (but see, e.g., [27, 26, 31, 39, 12, 16, 21] for software linguistics efforts at the SLE venue that occurred afterwards) and he suggests that we should use our own abstractions so that we feel inclined towards investing into usability. That’s exactly Eelco Visser’s approach; he is in a rather small, top league of building “stacks of DSLs for metaprogramming” to address, eventually, domains such as web programming.
 4. When being asked about “language entropy” (as I would like to call it now) and how to manage it (Q8 and Q9), Eelco doesn’t engage with the “philosophical” claim of mine that we may just have too many languages for no good reason. More pragmatically, he combines hopes in “standardization”, the “market”, and – quite insightfully – the ratio of effort to introduce and to maintain a language versus the productivity it enables.
 5. When being asked about his “pet kind of language” (Q10), Eelco refers to (languages for) the “easy implementation of IDEs” or what is best captured under the umbrella of language workbenches subsequently [9] (with the actual challenge being discussed by the communities as early as 2010); Eelco was very early in calling out this research direction.

Of course, Eelco Visser helped with launching the SLE conference not only by serving on the “formational” ATEM 2007 panel; in the process of setting up the SLE conference, he also advised on organizational aspects and community engineering.

<p>Q1: “My definition of SLE in elevator speech” <i>Building tools for building abstractions for improving programmer productivity.</i></p> <p>Q2: “SLE finally provides a home for xyz” <i>A systematic mapping of language engineering tradeoffs. [...]</i></p> <p>Q3: “An SLE challenge, readily waiting” <i>Safety of code generation. In the GPCE community, there is a lot of knowledge about safe code generation; syntax safety, type safety. In the industrial code generation community [...], code generation is done using text-based template engines.</i></p> <p>Q4: “Any useless kind of language in sight?” <i>Lots, but I don’t tend to remember these.</i></p> <p>Q5: “A praise on MDA as an SLE target” <i>For me, the contribution of MDA is that it has made high-level programming / modeling and code generation back on the agenda.</i></p> <p>Q6: “Clearance sale for programming languages” <i>Forget about MDA. Develop good run-time systems in a high-level PL. Build a stack of DSLs starting with technical domains, gradually covering more and more aspects of application domains.</i></p> <p>Q7: “Connections of SLE to linguistics et al.” <i>Linguistics: been there, not done that. Usability: yeah. That’s a general problem for language design. Basic approach: eat your own dog food (then you’ll work on usability). We/I don’t have the tools and time for doing large scale experiments.</i></p> <p>Q8: “Software language engineers to care for IT entropy” <i>Standardization is claimed to be an important tool to avoid entropy, but, realistically, what is the lifetime of programming paradigms/languages. Any program, in any language, will be legacy in a matter of years (say 5), [...].</i></p> <p>Q9: “If you were a congressman on planet HOPL ...” <i>Let’s quote (allegedly) Charles Simonyi: “The market will decide”. The big question: where is the tradeoff between effort required to implement a language and the size of the community of users that make this effort pay off. If it is very easy to define a language, a community size of 1 may be feasible (if making the language is cheaper than make the code that is generated from it). [...]</i></p> <p>Q10: “Specific engineering challenges for your pet kind of languages” <i>Easy implementation of IDEs for textual languages. Independent extensibility of transformations.</i></p>
--

■ **Figure 2** Eelco Visser’s notes for the ATEM 2007 panel – with some elisions (“[...]”).

2 Eelco Visser’s contributions to the SLE conference as an author

Here are two basic facts based on DBLP’s records for Eelco Visser and SLE 2008–2021:

- Eelco is the top-publishing SLE author: 18 papers.
- SLE is Eelco’s top-targeted venue.⁶

Despite these numbers, Eelco Visser should not be claimed by SLE. Eelco is an exceptional member of several other communities, which often also intersect with SLE – in particular GPCE, OOPSLA, IFIP WG 2.11, and IFIP WG 2.16.

⁶ DBLP lists 19 OOPSLA vs. 18 SLE papers, but several of the OOPSLA papers are extended abstracts.

Cite	Year	(Simplified) title	Syntax	Semantics	Typing	Transformation	IDE	Theory	Evaluation
[29]	2020	Gradually typing strategies	×	×	●	●	×	•	•
[6]	2018	Specification of indentation rules	●	×	×	×	×	•	●
[18]	2018	Migration to incremental computing	×	×	×	●	×	×	●
[8]	2018	Migration to a language workbench	×	×	×	•	●	×	•
[7]	2017	Priority conflicts	●	×	×	×	×	×	●
[28]	2017	Dataflow analysis specification	×	●	•	×	×	•	•
[5]	2016	Code completion	●	×	×	×	•	●	•
[17]	2014	Role-based data modeling and navigation	×	•	•	×	×	•	×
[9]	2013	Conclusions on the LWB challenge	•	•	•	•	•	×	●
[37]	2013	Incremental name and type analysis	•	●	●	×	●	×	●
[24]	2012	Name binding and scope rules	•	●	●	×	•	×	•
[4]	2011	Layout preservation	•	×	×	●	•	●	●
[32]	2011	Reconstruction of metamodel evolution	•	×	×	•	×	•	•
[22]	2010	Disambiguation of meta programs	●	×	×	•	•	•	•
[15]	2009	Data validation & UI concerns in WebDSL	•	×	•	•	×	×	×
[3]	2009	Parser error recovery	●	×	×	×	•	×	•
[20]	2009	Retargetable DSLs	×	•	×	●	×	×	•
[1]	2008	Parse table composition	●	×	×	×	×	•	•

■ **Figure 3** Eelco Visser’s SLE papers – with manually assigned codes.

This section though focuses on Eelco’s “SLE profile”. Fig. 3 lists Eelco’s SLE papers and attaches some info based on a simple coding scheme to organize these papers in terms of topic and methodology.⁷ In particular, I manually assigned codes for the following SLE-related *topics*:

Syntax Grammars (or metamodels or schemas) and their implementation.

Semantics Dynamic and translational semantics.

Typing Static semantics, type checking, and semantics analysis.

Transformation Program transformation and metaprogramming overall.

IDE IDE support and language workbenches.

Further, I manually assigned codes regarding *research methodology*:

Theory PL-like metatheory or other forms of formal analysis.

Evaluation Benchmarking or empirical study (e.g., a controlled experiment).

⁷ This section does not hold up to the standard of a systematic literature survey. The codes were extracted without any robust approach such as the systematic use of search strings. It helped though that the number of papers is manageable and the papers at hand are more or less familiar to me.

For each code, I manually assigned a *relevance* – to express how relevant the topic or the methodology is for the paper at hand:⁸

- × (Nearly) none
- Observable, but limited relevance
- Significant relevance
- Very significant relevance

The SLE “medley” of Fig. 3 gives rise to the following observations or claims:

1. Eelco’s contributions to SLE are concerned more often with syntax than with semantics, typing, or transformation. This is also due to the fact that some of the more semantical and transformational contributions went to other venues such as GPCE and OOPSLA. However, Eelco’s long-established interest in transformation/metaprogramming and his more recent engagement with semantic analysis also shows clearly at SLE.
2. The general trend – in the broader SE/PL communities – to require a proper form of validation shows in the increasing relevance of “Theory” and “Evaluation” in the table. In a few cases, the papers also “outsourced” validation to extra publications. Eelco focused on building tools and proving their utility; “Evaluation” is more common than “Theory”.
3. Despite the IDE topic being foreseen in 2007 (see Sec. 1), it took 6 years for the topic to materialize in Eelco’s publication record – at least as far as SLE is concerned, but see, of course, publications elsewhere on Spooifax from Eelco’s research team such as [23]. This is in part a testament to the speed of development of research topics, but also proof of other venues’ grab on thriving topics.
4. Despite “safety of code generation” being called out in 2007 as an SLE topic (see Sec. 1), Eelco preferred other venues for this topic (see, e.g., [33, 10, 11, 19]). When I started the coding scheme for Fig. 3, I expected to encounter the need for additional codes such as a potential code “code generation” as part of both “Semantics” and “Transformation”. However, the initial coding scheme was not ever refined because to be me it seemed that no additional (sub-) categories arose in the Eelco’s SLE portfolio.

Jean-Marie Favre – possibly the lead founder of SLE, definitely my key mentor regarding technological space travel and integration – told me around 10 years ago, in a discussion of “work-life balance”, that we are not the sum and not the count of our papers – certainly not so for our loved ones. Still for the “extended” SLE community, Eelco Visser’s publication record and the contributions he (with his many collaborators) made to the field are of a leadership significance – inside and outside the SLE scope.

3 Eelco Visser’s involvement in running the SLE conference

Eelco Visser helped with running the SLE conference over the years in several ways:

1. Chair SLE in 2021 [35].
2. Serve on the PC every now and then.
3. Submit to SLE like nobody else (see Sec. 2).
4. Support the co-location between GPCE and SLE.
5. Be available as a peer ever since launch (see Sec. 1).
6. Contribute to conference organization / online existence for SLE.
7. Attend and engage in meetings of the GPCE and SLE steering committees.

⁸ Please note that “relevance” of a SLE-related code requires that the paper aims at a contribution to the relevant “research field”. For instance, just because a paper includes a “grammar” for a DSL, this does not imply a relevance for the code “Syntax”, whereas a fundamental contribution to “parsing” would count as relevant.

From: Eelco Visser <...>
 Date: Fri, Oct 8, 2010 at 11:58 AM
 Subject: Re: Invitation to SLE 2011 Programme Committee Co-Chair
 To: Joao Saraiva <...>
 Cc: Ralf Lämmel <...>

Dear João,

Unfortunately I have to decline. I really like SLE, I'm honored by the invitation, and I would like to be PC chair, but the timing is not good. The timing of the conference itself is not so much the problem; I'll want to attend SLE anyway. But in addition to ICMT I just accepted to be co-chair of a workshop, and I'm in the OOPSLA'11 PC, which does overlap with the SLE reviewing period. Perhaps more importantly, several of my PhD students will be finishing their theses by next summer and I expect we'll want to submit lots of papers; SLE is an important outlet for us, and by being PC chair I would disadvantage my students by not being able to submit to SLE.

Anyway, I'm sorry I have to disappoint you, and I'll be available for the job in the future if the occasion would be present itself again.

cheers,
 – Eelco

■ **Figure 4** Eelco Visser's NO to the invitation of co-chairing the PC of SLE 2011.

Eelco eventually chaired the SLE conference in 2021, but he was surely asked more than once over the years. Much is known and documented about Eelco's skills to lead and to mentor; I want to include a more concealed entry here while covering an important skill nevertheless. That is, I was able to find one record where he kindly and thoughtfully declined; see Fig. 4. Here is what I call "**the discipline of declining**":

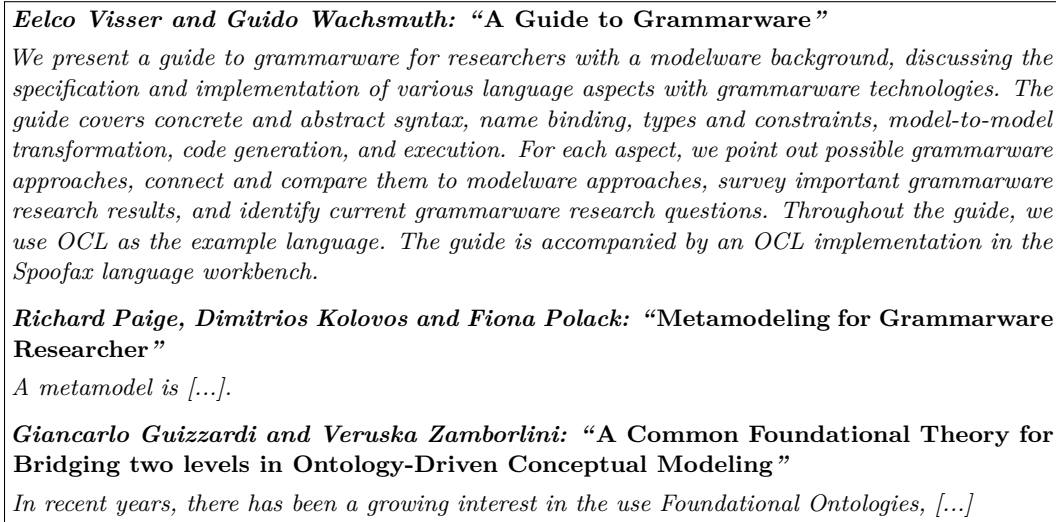
1. Cheer up the inquirers!
2. Don't accept a committee/chairing duty, if your schedule is too crowded already!
3. Make use of self-advertisement to point out what other important duties are in the way.
4. Optionally, tell them you are available in the future. Keep the door open, if you mean it.

4 Eelco Visser's contributions to SLEBoK

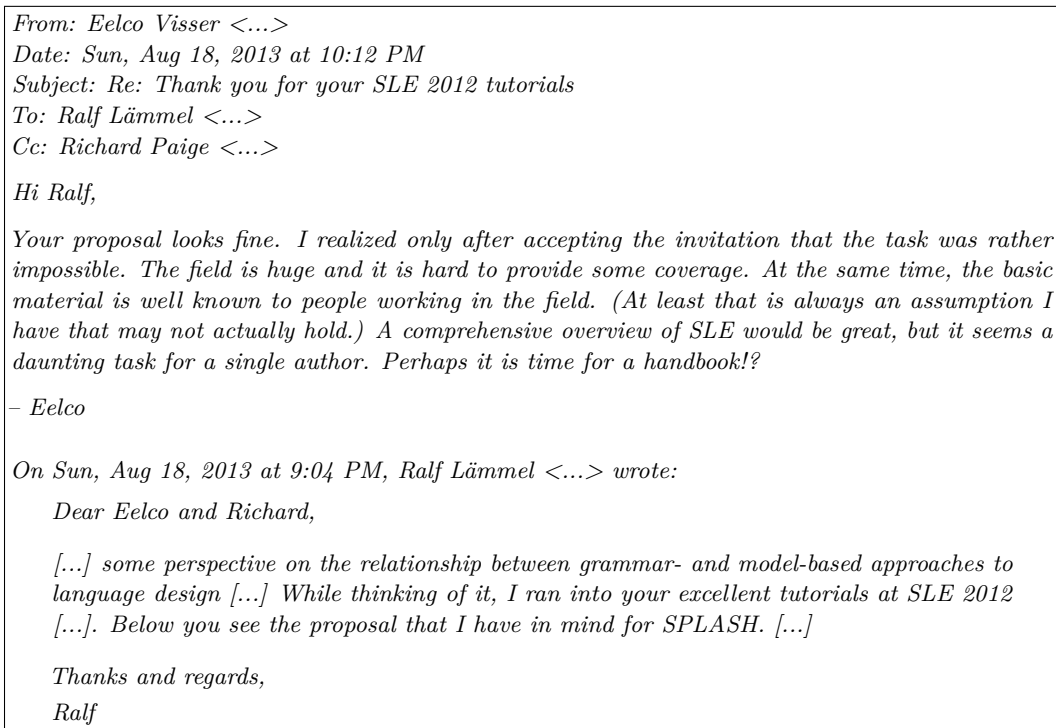
SLE, by its very definition [14] (also see the CFPs for SLE 2008–2022), aims at the integration of engineering knowledge regarding software languages across technological spaces, use cases, formalisms, etc.; see again Eelco's replies in Fig. 2 (Q1–Q3). Such knowledge integration was eventually understood as building a "*Body of Knowledge*" (BoK) – a notion used by various communities. Since 2012, we speak of the so-called SLEBoK – the *SLE Body of Knowledge*. Eelco Visser was involved in such activities explicitly aiming at knowledge integration in the SLE context, as highlighted below.

Along the lines of the SLEBoK idea, tutorial-style elements were exercised at SLE conferences or other venues (such as SPLASH) over the years. Notably, SLE 2012 in Dresden featured three mini-tutorials meant to support technological space travel – in the sense of explaining one space for an audience more settled in another space.⁹ In particular, Eelco Visser and Guido Wachsmuth contributed "*A Guide to Grammarware*"; see Fig. 5.

⁹ <https://www.sleconf.org/2012/Minitutorials.html>



■ **Figure 5** Minitutorials at SLE 2012 – including one by Eelco Visser.



■ **Figure 6** Eelco Visser suggesting there is a need for an SLE handbook.

I ran into those tutorials from SLE 2012 in Dresden only after the conference, when preparing my half-day SPLASH/SLE 2013 tutorial on “*Language Modeling Principles*”¹⁰ where I also tried to travel the spaces, as opposed to sticking to my comfort space (“grammarware”). My tutorial was the starting point for my work on the “Software Languages Book” [25]. I was encouraged by Eelco’s response regarding a draft proposal for the tutorial; see Fig. 6.

¹⁰<https://2013.splashcon.org/track/splash-2013-Tutorials>

From: Eelco Visser <...>
 Date: Mon, Mar 27, 2017 at 11:41 AM
 Subject: Re: Personal Invitation to Dagstuhl Seminar 17342
 To: Ralf Laemmel <...>, Eric Van Wyk <...>, Benoit Combemale <...>

Hi Ralf, Eric, Benoit,

Thanks for the invitation! Unfortunately I won't be able to attend the meeting, as I'll be traveling to a WG2.16 meeting that week.

I'm looking forward to the outcome of the meeting.

cheers,
 – Eelco

■ **Figure 7** Eelco Visser declining SLEBoK Dagstuhl attendance.

It so happens that SLE 2012 in Dresden was also the starting point for the actual SLEBoK label, subject to the SLEBoK workshop organized by Jean-Marie Favre and Jurgen Vinju.¹¹ Jean-Marie Favre also contributed some SLEBoK-like elements to the summer school SoTeSoLa (Software Technologies and Software Languages) in Koblenz, just a few months earlier.

In 2017, a representative portion of the SLE community met at Dagstuhl [2]¹² to properly get going on working out an actual SLEBoK-style collection of articles (a book for a BoK). At the Dagstuhl seminar, we had working groups on the following topics – with the names edited here for clarity:

- Reuse and modularity in specifications of software languages
- Attribute grammars
- A software language survey
- Parsing
- The SLE curriculum

The overall assumption is that such BoK-style efforts would support the continuous and well compartmentalized exchange between SLE researchers and practitioners (both of the “engineer” type). The SLEBoK Dagstuhl report [2] also features an opinion piece by Benoit Combemale “*On the need for a SLEBoK*”. Of course, Eelco Visser was invited to the Dagstuhl seminar, but he had to decline for good reasons; it is on the record that he was interested in the outcome; see Fig. 7.

Progress reports on SLEBoK were delivered at subsequent SLE conference editions; the effort stalled a bit due to the pandemic or otherwise. (Mea culpa!) The most recent conference edition (SLE 2022) encouraged SLEBoK-type submissions in the CFP, but only one such publication [30] (by Friedrich Steimann) was included. This publication provides, in my view, a very good example of a suitable format for SLEBoK papers and it will hopefully inspire others to follow.

In the meantime, textbooks such as [36, 25, 38]^{13,14,15} are of help, too. Perhaps not surprisingly, one of these books is coauthored by Eelco. Given my intimate knowledge of the other two books, I can say with certainty that they wouldn't exist without Eelco's contribution to the community, to its body of knowledge.

¹¹https://www.sleconf.org/2012/SLEBOK_SLE2012.html

¹²<https://www.dagstuhl.de/17342>

¹³<https://voelter.de/books.html>

¹⁴<http://www.softlang.org/book>

¹⁵<http://www.dsl.design>

5 Eelco Visser as the SLE photographer

This is a role never advertised and never formally claimed, but many of us will remember Eelco going around the conference venue and taking great shots. (He was doing this at all events he attended, not just at SLE!) This is another dimension of Eelco's community engagement which will be dearly missed.

<p><i>From: Ralf Lämmel <...></i> <i>Date: Fri, Apr 14, 2017 at 7:26 PM</i> <i>Subject: Permission for photo</i> <i>To: Eelco Visser <...></i></p> <p><i>Hi Eelco,</i></p> <p><i>may I kindly ask for permission to reuse this photo in my upcoming book?</i></p> <p><i>https://www.flickr.com/photos/eelcovisser/4772847104</i></p> <p><i>Thank you,</i> <i>Ralf</i></p>
<p><i>From: Eelco Visser <...></i> <i>Date: Fri, Apr 14, 2017 at 7:38 PM</i> <i>Subject: Re: Permission for photo</i> <i>To: Ralf Lämmel <...></i></p> <p><i>Sure, with full credits and if Jean-Marie agrees as well.</i> <i>[...]</i> <i>- Eelco</i></p>

■ **Figure 8** Eelco Visser giving permission to use his 2012 photo of Jean-Marie Favre.

While completing the “Software Languages Book” [25], I ran into a photo of Jean-Marie Favre taken by Eelco, which I wanted to artistically transform with a deep-learning approach for inclusion in the book; see Fig. 8 for the conversation; see the book for a transformed Jean-Marie Favre based on Eelco's photo in Chapter 1; the artwork is also available online.¹⁶



■ **Figure 9** A photo of Eelco in photographer mode at SLE 2008.

¹⁶<http://softlang.uni-koblenz.de/book/artwork.pdf>

I want to close these notes in a reasonably “meta” manner – with a (sentence about a) (a reproduction of a) *photo I took of Eelco while he was taking a photo* of something going on at SLE 2008 in Toulouse; see Fig. 9. (In this manner, I also want to allude to the meta-meta focus in Eelco’s research: languages of languages.¹⁷)

References

- 1 Martin Bravenboer and Eelco Visser. Parse table composition. In Dragan Gasevic, Ralf Lämmel, and Eric Van Wyk, editors, *Software Language Engineering, First International Conference, SLE 2008, Toulouse, France, September 29-30, 2008. Revised Selected Papers*, volume 5452 of *Lecture Notes in Computer Science*, pages 74–94. Springer, 2008. doi:10.1007/978-3-642-00434-6_6.
- 2 Benoît Combemale, Ralf Lämmel, and Eric Van Wyk. SLEBOK: the software language engineering body of knowledge (dagstuhl seminar 17342). *Dagstuhl Reports*, 7(8):45–54, 2017. doi:10.4230/DagRep.7.8.45.
- 3 Maartje de Jonge, Emma Nilsson-Nyman, Lennart C. L. Kats, and Eelco Visser. Natural and flexible error recovery for generated parsers. In Mark van den Brand, Dragan Gasevic, and Jeff Gray, editors, *Software Language Engineering, Second International Conference, SLE 2009, Denver, CO, USA, October 5-6, 2009, Revised Selected Papers*, volume 5969 of *Lecture Notes in Computer Science*, pages 204–223. Springer, 2009. doi:10.1007/978-3-642-12107-4_16.
- 4 Maartje de Jonge and Eelco Visser. An algorithm for layout preservation in refactoring transformations. In Anthony M. Sloane and Uwe Aßmann, editors, *Software Language Engineering - 4th International Conference, SLE 2011, Braga, Portugal, July 3-4, 2011, Revised Selected Papers*, volume 6940 of *Lecture Notes in Computer Science*, pages 40–59. Springer, 2011. doi:10.1007/978-3-642-28830-2_3.
- 5 Luís Eduardo de Souza Amorim, Sebastian Erdweg, Guido Wachsmuth, and Eelco Visser. Principled syntactic code completion using placeholders. In Tijs van der Storm, Emilie Balland, and Dániel Varró, editors, *Proceedings of the 2016 ACM SIGPLAN International Conference on Software Language Engineering, Amsterdam, The Netherlands, October 31 - November 1, 2016*, pages 163–175. ACM, 2016. URL: <http://dl.acm.org/citation.cfm?id=2997374>.
- 6 Luís Eduardo de Souza Amorim, Michael J. Steindorfer, Sebastian Erdweg, and Eelco Visser. Declarative specification of indentation rules: a tooling perspective on parsing and pretty-printing layout-sensitive languages. In David J. Pearce, Tanja Mayerhofer, and Friedrich Steimann, editors, *Proceedings of the 11th ACM SIGPLAN International Conference on Software Language Engineering, SLE 2018, Boston, MA, USA, November 05-06, 2018*, pages 3–15. ACM, 2018. doi:10.1145/3276604.3276607.
- 7 Luís Eduardo de Souza Amorim, Michael J. Steindorfer, and Eelco Visser. Deep priority conflicts in the wild: a pilot study. In Benoît Combemale, Marjan Mernik, and Bernhard Rumpe, editors, *Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering, SLE 2017, Vancouver, BC, Canada, October 23-24, 2017*, pages 55–66. ACM, 2017. doi:10.1145/3136014.3136020.
- 8 Jasper Denkers, Louis van Gool, and Eelco Visser. Migrating custom DSL implementations to a language workbench (tool demo). In David J. Pearce, Tanja Mayerhofer, and Friedrich Steimann, editors, *Proceedings of the 11th ACM SIGPLAN International Conference on Software Language Engineering, SLE 2018, Boston, MA, USA, November 05-06, 2018*, pages 205–209. ACM, 2018. doi:10.1145/3276604.3276608.
- 9 Sebastian Erdweg, Tijs van der Storm, Markus Völter, Meinte Boersma, Remi Bosman, William R. Cook, Albert Gerritsen, Angelo Hulshout, Steven Kelly, Alex Loh, Gabriël D. P. Konat, Pedro J. Molina, Martin Palatnik, Risto Pohjonen, Eugen Schindler, Klemens Schindler,

¹⁷Thanks to Friedrich Steimann who pointed out some possible transformations of the sentence such that a meta-circular or meta-recursive situation may arise. This requires more reflection on my side.

- Riccardo Solmi, Vlad A. Vergu, Eelco Visser, Kevin van der Vlist, Guido Wachsmuth, and Jimi van der Woning. The state of the art in language workbenches - conclusions from the language workbench challenge. In Martin Erwig, Richard F. Paige, and Eric Van Wyk, editors, *Software Language Engineering - 6th International Conference, SLE 2013, Indianapolis, IN, USA, October 26-28, 2013. Proceedings*, volume 8225 of *Lecture Notes in Computer Science*, pages 197–217. Springer, 2013. doi:10.1007/978-3-319-02654-1_11.
- 10 Sebastian Erdweg, Vlad A. Vergu, Mira Mezini, and Eelco Visser. Finding bugs in program generators by dynamic analysis of syntactic language constraints. In Walter Binder, Erik Ernst, Achille Peternier, and Robert Hirschfeld, editors, *13th International Conference on Modularity, MODULARITY '14, Lugano, Switzerland, April 22-26, 2014*, pages 17–20. ACM, 2014. doi:10.1145/2584469.2584474.
 - 11 Sebastian Erdweg, Vlad A. Vergu, Mira Mezini, and Eelco Visser. Modular specification and dynamic enforcement of syntactic language constraints when generating code. In Walter Binder, Erik Ernst, Achille Peternier, and Robert Hirschfeld, editors, *13th International Conference on Modularity, MODULARITY '14, Lugano, Switzerland, April 22-26, 2014*, pages 241–252. ACM, 2014. doi:10.1145/2577080.2577089.
 - 12 Jean-Marie Favre, Dragan Gasevic, Ralf Lämmel, and Ekaterina Pek. Empirical language analysis in software linguistics. In Brian A. Malloy, Steffen Staab, and Mark van den Brand, editors, *Software Language Engineering - Third International Conference, SLE 2010, Eindhoven, The Netherlands, October 12-13, 2010, Revised Selected Papers*, volume 6563 of *Lecture Notes in Computer Science*, pages 316–326. Springer, 2010. doi:10.1007/978-3-642-19440-5_21.
 - 13 Jean-Marie Favre, Dragan Gasevic, Ralf Lämmel, and Andreas Winter. 4th international workshop on language engineering (ATEM 2007). In Holger Giese, editor, *Models in Software Engineering, Workshops and Symposia at MoDELS 2007, Nashville, TN, USA, September 30 - October 5, 2007, Reports and Revised Selected Papers*, volume 5002 of *Lecture Notes in Computer Science*, pages 28–33. Springer, 2007. doi:10.1007/978-3-540-69073-3_4.
 - 14 Jean-Marie Favre, Dragan Gasevic, Ralf Lämmel, and Andreas Winter. Guest editors' introduction to the special section on software language engineering. *IEEE Trans. Software Eng.*, 35(6):737–741, 2009. doi:10.1109/TSE.2009.78.
 - 15 Danny M. Groenewegen and Eelco Visser. Integration of data validation and user interface concerns in a DSL for web applications. In Mark van den Brand, Dragan Gasevic, and Jeff Gray, editors, *Software Language Engineering, Second International Conference, SLE 2009, Denver, CO, USA, October 5-6, 2009, Revised Selected Papers*, volume 5969 of *Lecture Notes in Computer Science*, pages 164–173. Springer, 2009. doi:10.1007/978-3-642-12107-4_13.
 - 16 Jurriaan Hage and Peter van Keeken. Neon: A library for language usage analysis. In Dragan Gasevic, Ralf Lämmel, and Eric Van Wyk, editors, *Software Language Engineering, First International Conference, SLE 2008, Toulouse, France, September 29-30, 2008. Revised Selected Papers*, volume 5452 of *Lecture Notes in Computer Science*, pages 35–53. Springer, 2008. doi:10.1007/978-3-642-00434-6_4.
 - 17 Daco Harkes and Eelco Visser. Unifying and generalizing relations in role-based data modeling and navigation. In Benoît Combemale, David J. Pearce, Olivier Barais, and Jurgen J. Vinju, editors, *Software Language Engineering - 7th International Conference, SLE 2014, Västerås, Sweden, September 15-16, 2014. Proceedings*, volume 8706 of *Lecture Notes in Computer Science*, pages 241–260. Springer, 2014. doi:10.1007/978-3-319-11245-9_14.
 - 18 Daco C. Harkes, Elmer van Chastelet, and Eelco Visser. Migrating business logic to an incremental computing DSL: a case study. In David J. Pearce, Tanja Mayerhofer, and Friedrich Steimann, editors, *Proceedings of the 11th ACM SIGPLAN International Conference on Software Language Engineering, SLE 2018, Boston, MA, USA, November 05-06, 2018*, pages 83–96. ACM, 2018. doi:10.1145/3276604.3276617.
 - 19 Zef Hemel, Lennart C. L. Kats, Danny M. Groenewegen, and Eelco Visser. Code generation by model transformation: a case study in transformation modularity. *Softw. Syst. Model.*, 9(3):375–402, 2010. doi:10.1007/s10270-009-0136-1.

- 20 Zef Hemel and Eelco Visser. PIL: A platform independent language for retargetable dsls. In Mark van den Brand, Dragan Gasevic, and Jeff Gray, editors, *Software Language Engineering, Second International Conference, SLE 2009, Denver, CO, USA, October 5-6, 2009, Revised Selected Papers*, volume 5969 of *Lecture Notes in Computer Science*, pages 224–243. Springer, 2009. doi:10.1007/978-3-642-12107-4_17.
- 21 Einar W. Høst and Bjarte M. Østvold. The java programmer’s phrase book. In Dragan Gasevic, Ralf Lämmel, and Eric Van Wyk, editors, *Software Language Engineering, First International Conference, SLE 2008, Toulouse, France, September 29-30, 2008. Revised Selected Papers*, volume 5452 of *Lecture Notes in Computer Science*, pages 322–341. Springer, 2008. doi:10.1007/978-3-642-00434-6_20.
- 22 Lennart C. L. Kats, Karl Trygve Kalleberg, and Eelco Visser. Interactive disambiguation of meta programs with concrete object syntax. In Brian A. Malloy, Steffen Staab, and Mark van den Brand, editors, *Software Language Engineering - Third International Conference, SLE 2010, Eindhoven, The Netherlands, October 12-13, 2010, Revised Selected Papers*, volume 6563 of *Lecture Notes in Computer Science*, pages 327–336. Springer, 2010. doi:10.1007/978-3-642-19440-5_22.
- 23 Lennart C. L. Kats and Eelco Visser. The spoofax language workbench: rules for declarative specification of languages and ides. In William R. Cook, Siobhán Clarke, and Martin C. Rinard, editors, *Proceedings of the 25th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2010, October 17-21, 2010, Reno/Tahoe, Nevada, USA*, pages 444–463. ACM, 2010. doi:10.1145/1869459.1869497.
- 24 Gabriël D. P. Konat, Lennart C. L. Kats, Guido Wachsmuth, and Eelco Visser. Declarative name binding and scope rules. In Krzysztof Czarnecki and Görel Hedin, editors, *Software Language Engineering, 5th International Conference, SLE 2012, Dresden, Germany, September 26-28, 2012, Revised Selected Papers*, volume 7745 of *Lecture Notes in Computer Science*, pages 311–331. Springer, 2012. doi:10.1007/978-3-642-36089-3_18.
- 25 Ralf Lämmel. *Software Languages: Syntax, Semantics, and Metaprogramming*. Springer, 2018.
- 26 Suman Roychoudhury, Sagar Sunkle, Deepali Kholkar, and Vinay Kulkarni. A domain-specific controlled english language for automated regulatory compliance (industrial paper). In Benoît Combemale, Marjan Mernik, and Bernhard Rumpe, editors, *Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering, SLE 2017, Vancouver, BC, Canada, October 23-24, 2017*, pages 175–181. ACM, 2017. doi:10.1145/3136014.3136018.
- 27 Philipp Seifer, Johannes Härtel, Martin Leinberger, Ralf Lämmel, and Steffen Staab. Empirical study on the usage of graph query languages in open source java projects. In Oscar Nierstrasz, Jeff Gray, and Bruno C. d. S. Oliveira, editors, *Proceedings of the 12th ACM SIGPLAN International Conference on Software Language Engineering, SLE 2019, Athens, Greece, October 20-22, 2019*, pages 152–166. ACM, 2019. doi:10.1145/3357766.3359541.
- 28 Jeff Smits and Eelco Visser. Flowspec: declarative dataflow analysis specification. In Benoît Combemale, Marjan Mernik, and Bernhard Rumpe, editors, *Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering, SLE 2017, Vancouver, BC, Canada, October 23-24, 2017*, pages 221–231. ACM, 2017. doi:10.1145/3136014.3136029.
- 29 Jeff Smits and Eelco Visser. Gradually typing strategies. In Ralf Lämmel, Laurence Tratt, and Juan de Lara, editors, *Proceedings of the 13th ACM SIGPLAN International Conference on Software Language Engineering, SLE 2020, Virtual Event, USA, November 16-17, 2020*, pages 1–15. ACM, 2020. doi:10.1145/3426425.3426928.
- 30 Friedrich Steimann and Marius Freitag. The semantics of plurals. In *Software Language Engineering, International Conference, SLE 2022*. ACM DL, 2022.
- 31 Eric Umuhoza, Marco Brambilla, Davide Ripamonti, and Jordi Cabot. An empirical study on simplification of business process modeling languages. In Richard F. Paige, Davide Di Ruscio, and Markus Völter, editors, *Proceedings of the 2015 ACM SIGPLAN International Conference on Software Language Engineering, SLE 2015, Pittsburgh, PA, USA, October 25-27, 2015*, pages 13–24. ACM, 2015. doi:10.1145/2814251.2814254.

- 32 Sander Vermolen, Guido Wachsmuth, and Eelco Visser. Reconstructing complex metamodel evolution. In Anthony M. Sloane and Uwe Aßmann, editors, *Software Language Engineering - 4th International Conference, SLE 2011, Braga, Portugal, July 3-4, 2011, Revised Selected Papers*, volume 6940 of *Lecture Notes in Computer Science*, pages 201–221. Springer, 2011. doi:10.1007/978-3-642-28830-2_11.
- 33 Sander Daniël Vermolen, Guido Wachsmuth, and Eelco Visser. Generating database migrations for evolving web applications. In Ewen Denney and Ulrik Pagh Schultz, editors, *Generative Programming And Component Engineering, Proceedings of the 10th International Conference on Generative Programming and Component Engineering, GPCE 2011, Portland, Oregon, USA, October 22-24, 2011*, pages 83–92. ACM, 2011. doi:10.1145/2047862.2047876.
- 34 Eelco Visser, Zine-El-Abidine Benaïssa, and Andrew P. Tolmach. Building program optimizers with rewriting strategies. In Matthias Felleisen, Paul Hudak, and Christian Queinnec, editors, *Proceedings of the third ACM SIGPLAN International Conference on Functional Programming (ICFP '98), Baltimore, Maryland, USA, September 27-29, 1998*, pages 13–26. ACM, 1998. doi:10.1145/289423.289425.
- 35 Eelco Visser, Dimitris S. Kolovos, and Emma Söderberg, editors. *SLE '21: 14th ACM SIGPLAN International Conference on Software Language Engineering, Chicago, IL, USA, October 17 - 18, 2021*. ACM, 2021. doi:10.1145/3486608.
- 36 Markus Voelter. *DSL Engineering. Designing, Implementing and Using Domain-Specific Languages*. CreateSpace Independent Publishing Platform, 2013.
- 37 Guido Wachsmuth, Gabriël D. P. Konat, Vlad A. Vergu, Danny M. Groenewegen, and Eelco Visser. A language independent task engine for incremental name and type analysis. In Martin Erwig, Richard F. Paige, and Eric Van Wyk, editors, *Software Language Engineering - 6th International Conference, SLE 2013, Indianapolis, IN, USA, October 26-28, 2013. Proceedings*, volume 8225 of *Lecture Notes in Computer Science*, pages 260–280. Springer, 2013. doi:10.1007/978-3-319-02654-1_15.
- 38 Andrzej Wasowski and Thorsten Berger. *Domain-Specific Languages: Effective modeling, automation, and reuse*. Springer, 2023.
- 39 Vadim Zaytsev and Ralf Lämmel. A unified format for language documents. In Brian A. Malloy, Steffen Staab, and Mark van den Brand, editors, *Software Language Engineering - Third International Conference, SLE 2010, Eindhoven, The Netherlands, October 12-13, 2010, Revised Selected Papers*, volume 6563 of *Lecture Notes in Computer Science*, pages 206–225. Springer, 2010. doi:10.1007/978-3-642-19440-5_13.