Explaining Eye Movements in Program Comprehension using jACT-R

Sebastian Lohmeier (sl@monochromata.de) Nele Russwinkel (nele.russwinkel@tu-berlin.de)

Keywords: Eye Movements, Program Comprehenion, Tools, Java, Cohesion, Activation

Technische Universität Berlin

Model Input from Program Comprehension

- Available knowledge representations are insufficient when submitting real-world texts to cognitive models of text comprehension.
- These models have also been used to explain program comprehension experiments (Burkhardt, Détienne, & Wiedenbeck, 1997).
- To overcome the lack of representations, we aim at using source code as a knowledge representation to model program comprehension.
- Hansen, Lumsdaine, and Goldstone (2012) proposed a goal-directed cognitive model of program comprehension.
- They aimed at exact generative models of programming steps, while we seek a rough reconstructive model of the comprehension of large amounts of source code.
- Low activation of knowledge representations in our model could explain regressive eye movements that indicate comprehension problems.
- We track the eye movements of programmers while they read source code.
- Afterwards, chunks of conceptual knowledge are extracted from the source code.
- The model maintains an on-line representation of meaning by instantiating nodes from conceptual chunks and by re-activating existing nodes.
- Activation values are computed for conceptual knowledge and nodes of on-line meaning.

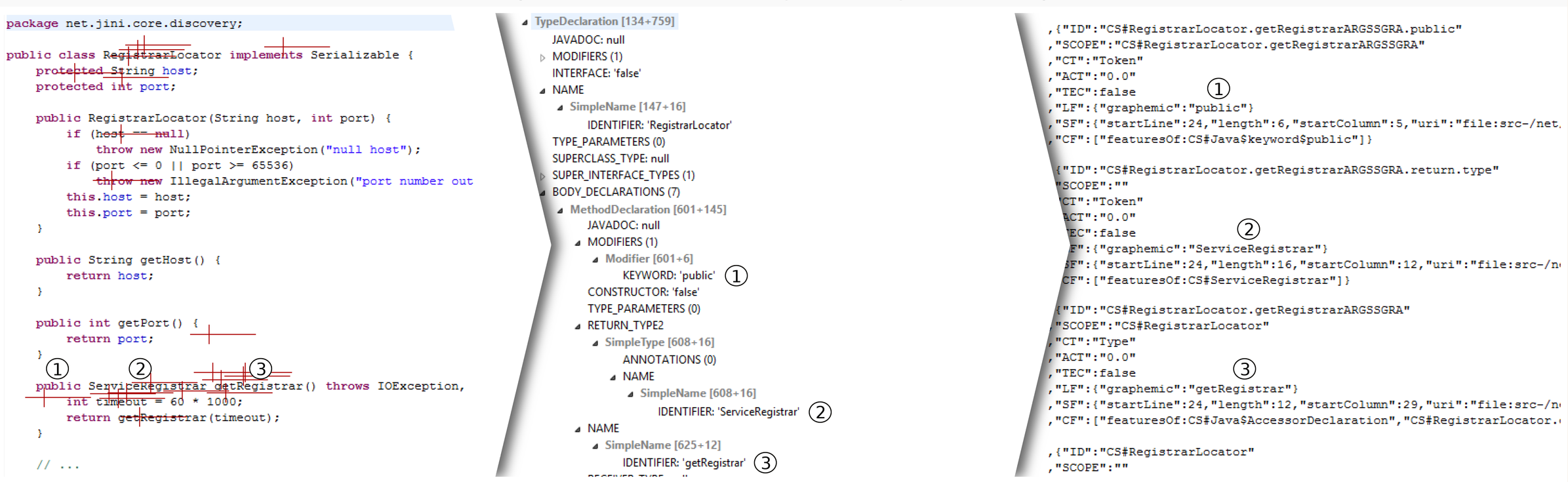

Figure 1: Sample eye movements on source code, a simplified abstract syntax tree (AST) of the code, and chunks in declarative memory generated from the AST

The input to the model is generated as follows.

- Subjects' fixations are mapped to words in source code using a plugin for the Eclipse IDE.
- A log containing fixations (incl. word, duration and location) and saccades (incl. duration) is created.
- The Java parser creates an abstract syntax tree (AST) when a file of source code is opened.
- The AST provides access to all concepts and relations expressed in the code, e.g. part-whole, subconcept-of, and is-a.
- Concepts are turned into type chunks, relations and objects are turned into token chunks.
- Reference potentials are created that combine a word with an optional link to a type chunk and spatial information that fixations can be assigned to.

Explaining Program Comprehension with jACT-R

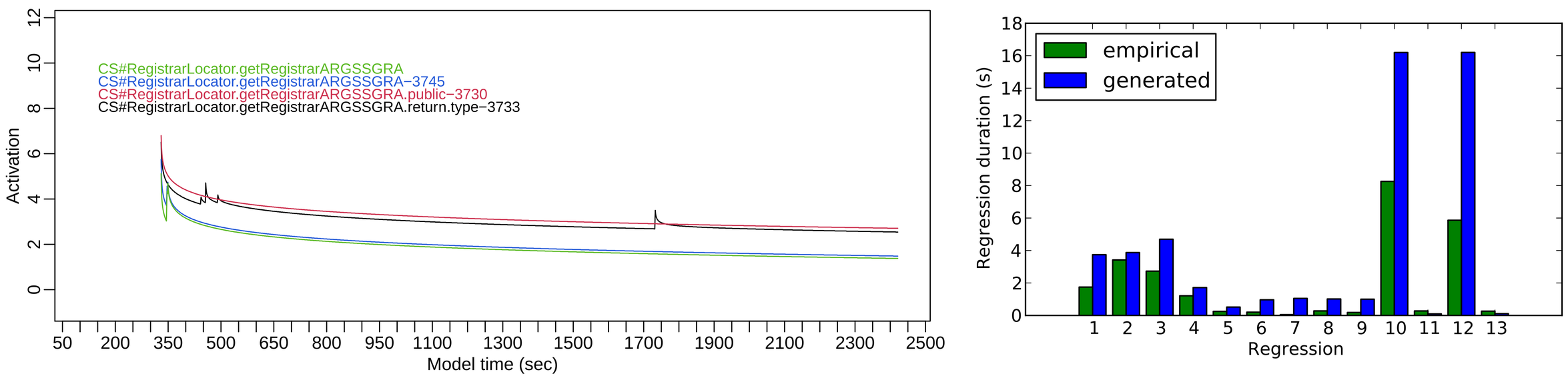

Figure 2: Sample time course of activation for chunks from Figure 1 (left) and empirical vs. model-generated regression durations (right)

- A cognitive model is constructed using jACT-R (`http://jact-r.org/`), a re-implementation of ACT-R (Anderson et al., 2004) written in Java.
- The log of fixations and saccades is read by a REMMA module that re-generates fixation durations following Salvucci (2001). The module creates reference potentials for fixated words.
- The reference potentials encoded by the REMMA module are used by a second module to instantiate new tokens from type nodes and to re-activating existing token nodes.
- Figure 2 details a model run using the input from Figure 1. The black line (left) shows the activation of the token node that represents the return type of the getRegistrar() method. This token is the first node retrieved during regression 1 on the right of Figure 2.
- The model over-estimates regression durations.
- Retrieval errors and retrieval duration of the token node referred to at the start of the regression does not explain regressions. Regression paths need to be examined closer.
- The model is still in an early stage. From 40 minutes of eye movements it creates 15,000 chunks using 9 chunk types and 2 productions.
- Base-level activation and spreading activation in the model need to be adjusted further.
- We are interested in predicting comprehension difficulties based on current activation and decay until a future point in time.
- Being based on the Eclipse IDE, the model potentially lends itself to interactive applications of cognitive modelling.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of mind. *Psychological Review*, *111*(4), 1036–1060.

Burkhardt, J.-M., Détienne, F., & Wiedenbeck, S. (1997). Mental representations constructed by experts and novices in object-oriented program comprehension. In S. Howard, J. Hammond, & G. Lingaard (Eds.), *Human-computer interaction: INTERACT '97.* London: Chapman & Hall.

Hansen, M. E., Lumsdaine, A., & Goldstone, R. L. (2012). Cognitive architectures: A way forward for the psychology of programming. In *Onward! 2012: Proceedings of the ACM international symposium on New ideas, new paradigms, and reflections on programming and software* (pp. 27–37).

Salvucci, D. D. (2001). An integrated model of eye movements and visual encoding. *Cognitive Systems Research*, *1*(4), 201–220.

Acknowledgements and Authors

We thank Anthony Harrison for implementing jACT-R and for helping us to get started with it.

Sebastian Lohmeier
sl@monochromata.de
TU Berlin
MSc Informatik
degree course

Nele Russwinkel
nele.russwinkel
@tu-berlin.de
TU Berlin
FG KMoDyS