

Implementing Object Tracking in the Emerald Object-Oriented Language

Eric Jul

University of Oslo, Norway

ABSTRACT This article offers an overview of the invited talk presented by Eric Jul at IC00OLPS 2025.

KEYWORDS Emerald, Object Tracking, Object-Oriented Programming.

Summary

In this talk, a language modification to Emerald is presented that allows objects to be tracked when they move around. First, the Emerald language is presented including its conformity based type system. Then the proposed language construct to enable objects to be tracked is presented. We introduce a new statement: *"track A using B"* whereafter the object referenced by A is tracked by B where B is a track handler, which is an object that contains a track handler operation named `objectMoved` that the underlying run-time system will call every time A moves to a different machine (called nodes in Emerald).

```
operation objectMoved[obj: Any, newNode: Node, arrivalTime: Time, moveCount: Integer]
```

In this way, object B can keep track of the object `obj` every move to a different node (`newNode`) and the sequence number of the move (`moveCount`) and the local time on the new Node.

Our implementation has shown some important performance characteristics: First, non-tracked object that move have no performance degradation, second, the performance impact of tracked objects that move is on the same order of magnitude as the move itself, i.e., typically the invocation of `objectMoved` is a remote call costing two network messages (if the tracking object is on the same Node as the object has moved to then, obviously, this is merely a very fast local call).

JOT reference format:

Eric Jul. *Implementing Object Tracking in the Emerald Object-Oriented Language*. Journal of Object Technology. Vol. 25, No. 1, 2026. Licensed under Attribution 4.0 International (CC BY 4.0)
<http://dx.doi.org/10.5381/jot.2026.25.1.a12>

We conclude that our suggested solution to the tracking problem is clean because it is at the language level (rather than a "hack" to a run-time system) and that the performance is as good as possible.

Acknowledgments

Master student at UiO Emir Kukuruzovic implemented the tracking mechanism in Emerald as part of his Masters thesis.

About the Presenter

Eric Jul has a Ph.D. in Computer Science from the University of Washington in Seattle, Washington, USA, where he was part of a small group developing the Distributed Programming Language Emerald. Eric was the principal implementor of the run-time system that supported fine-grained object to move within a distributed system including any threads executing inside the objects. From 1988 to 2009, he was at the Department of Computer Science (DIKU), University of Copenhagen, where he was head of the Distlab Group, and, from 1999, full professor. He was a member of Technical Staff at Bell Labs, Dublin 2009-2015. During that time he was Adjunct Professor at Trinity College Dublin. From 2009 he was a Professor II (a part-time professor) at the University of Oslo until he became a full professor there in the Object Modelling and Systems Group, which eventually became the Programming Technology Group. Professor Jul retired from the University of Oslo at the end of September 2025. He is now Research Director at a startup in Oslo, `newcode.ai`, that provides AI-based work support especially for Law Firms. You can contact the author at ericbj@uio.no.