

## Barbara Liskov

As a clear pioneer in the field of computer science, Barbara Liskov has made contributions to fields ranging from “practical and theoretical foundations of programming language and system design, especially related to data abstraction, fault tolerance, and distributed computing” (Vleck, 2019). Born in California as Barbara Huberman, Liskov hails from the prestigious school of the University of California, Berkeley where she received a Bachelor’s in Mathematics. She kickstarted her career at a company named Mitre before quickly leaving to do research and development at both Harvard and Stanford. She eventually became one of the first women to earn a PhD in Computer Science by conducting a graduate thesis on chess endgames.

After her time at Stanford, Barbara married a man named Nathan Liskov and moved back to Boston and went back to work at Mitre. It was there that she created an operating system called Venus, a small timesharing system used to experiment with how different architectures help or hinder the aforementioned process. In 1971, Liskov went to MIT as a faculty member to focus on creating more reliable computer systems. She led the development of the CLU language, building the foundations of many concepts later used in high level languages such as Java and C#. At MIT, she also created Argus, an experimental language which eased implementation of programs distributed over a network.

She continued to research in the area of distribution systems, covering “many aspects of operating systems and computation, including important work on object-oriented database systems, garbage collection, caching, persistence, recovery, fault tolerance, security, decentralized information flow, modular upgrading of distributed systems, geographic routing, and practical Byzantine fault tolerance” (Vleck, 2019). She, working with a woman named Jeannette Wing, developed the Liskov substitution principle, a new notion of subtyping which

influenced advanced system developments and set a standard for clarity and usefulness. Aged 81, she is now the author of over one hundred technical papers and three books.

In 2003, Liskov wrote a paper on a new application called MapJax titled: “MapJAX: Data Structure Abstractions for Asynchronous Web Applications”. She starts off the paper by describing AJAX, a program developed to send requests to a given server using Javascript without having to reload the page or block the user interface. She states that MapJax, which is “a data-centric framework for building AJAX applications without any new client-side software (Myers, 2007), was developed to be a mapping application to store URLs which correspond to maps. Every MapJAX program consists of a number of threads, which are created each time an event handler gets invoked by the browser. The MapJAX program also sports a prefetching feature so, when MapJAX is given a key, it will return multiple other keys which identify elements to be prefetched. However, MapJAX is somewhat bandwidth intensive, and thus a technique called request cancelling was implemented. When the application determines that certain elements are no longer required for the program to run successfully, it will simply cancel the request for it. Liskov and her team write “A major goal for MapJAX is to allow programmers to access data at servers using normal method calls, thus avoiding the complexity of programming with continuations and callbacks.”

The paper goes on to talk about different implementations of MapJAX such as the AutoComplete function, the Mapping implementation, and the WebMail application before going into experimental runtimes and latencies of applications. The three tested cases were as follows: without MapJAX, with MapJAX but without prefetching, and with MapJAX with prefetching. At networks configurations with relatively low bandwidths, using prefetching heavily increased the time that it took to use the program with prefetching, but for bandwidths

higher than 256Kb, the time taken was heavily reduced when prefetching was introduced.

Interestingly, latencies actually increased with MapJAX but without prefetching in every application, albeit not by much. It seems that there is never any benefit to using MapJAX without prefetching as opposed to using any other non-MapJAX program, in terms of runtime at least.

### Annotated Bibliography

Vleck, Tom Van. 2019. *Barbara Liskov - A.M. Turing Award Laureate*, ACM Turing Award.

Retrieved January 19, 2021 from [amturing.acm.org/award\\_winners/liskov\\_1108679.cfm](http://amturing.acm.org/award_winners/liskov_1108679.cfm).

Myers, Daniel, et al. MIT, 2007, pp. 1–14, MapJAX: Data Structure Abstractions for Asynchronous Web Applications.