

Article

[Athanasios Hatzis](#) · Sep 12, 2017 3m read

The wizards of stored computer program and the next generation of programmers



Introduction

Have you noticed that what ever the model and data structure in databases we cannot escape from the fundamental principle of managing data allocation space with references, i.e. [pointer](#) based logic, memory addressing ? Isn ' t this the fundamental mechanism of programming languages too ? The problem I see with all these modern NoSQL databases, especially graph databases is that they provide a higher level abstraction for the end developer but they hide and lock completely the access to the low level storage and retrieval mechanism including indexes. Even in key-value stores you cannot see or understand the sorting of indexes, you cannot easily reference data values in permanent storage locations.

Transparent operations

Transparency in computer science is a huge issue. The wizards and pioneers of computer hardware and software, have created multiple abstraction layers. They have hidden a lot of detail but at the same time they prevented access to low level mechanisms. And here comes the next generation that is asked to program the machine without understanding what is going on underneath. And even if there is such a desire, from the developer ' s point of view, the operating system environment, the language and the debugging tools, not to mention the market, do not help us towards this direction. There have been few exceptions and one of them has been MUMPS ([Massachusetts General Hospital Utility Multi-Programming System](#)). I found astonishing the fact that M was developed by Neil Pappalardo, Robert Greenes, and Curt Marble during 1966 and 1967. A reputable and established company in operational database market, Intersystems Cache, has taken over, expanded, and continued their effort on our days.

A programming language with a built-in database

[Cache ObjectScript](#) does make the difference from many aspects. Here I would like to emphasize its strong characteristic of a built-in database with sub-scripted arrays and multi-dimensional keys similar to the variables used by most programming languages to access main memory. Let me repeat this, you get a programming language with a built-in [multi-model database](#). I think this is a fundamental aspect that software pioneers have been missing when they invented new programming languages or new sql databases. They are missing the innate

common characteristic that both databases and programming languages share, which is the pointer, reference based logic. What do you think ?

Epilogue

So I believe it ' s time to return back and fix this for new generation multi-model databases AND post-modern programming languages too. We have already made a good start and we believe M will make history again in database management and modeling with our [Associative Semiotic Hypergraph Engine](#) that will be built on top of InterSystems Cache powerful [global data structure](#) and accompanied with a popular and easy to program client API in Python for data analytics.

We build powerful meaningful relationships easily

© HEALIS - Athanassios I. Hatzis, April 2017

[#Databases](#) [#Data Model](#) [#Globals](#) [#Cache](#)

Source

URL:<https://community.intersystems.com/post/wizards-stored-computer-program-and-next-generation-programmers>