

SHORT

About Arjen Veneman

Arjen Veneman is founder and co-owner of Boon Software. He has 15 years of experience in building e-learning frameworks and solutions. Since 2012 we gradually started making the shift from as3/php to haxe and neko. Today we have a framework build with Haxe upon we create custom made e-learning solutions.

About the talk: “Using haxe to build an e-learning framework”

In the talk I'll discuss how we used haxe to create a model-driven, future proof framework. Our framework is build with haxe and allows us to create custom made e-learning solutions.

I like to discuss 3 main aspects of this framework and how haxe helped us establishing them.

1. Model driven architecture
2. High structural and visual flexibility
3. Minimal language (haxe-target) dependencies

LONG

About Boon

Boon is an e-learning technology company. We're building e-learning software for 15 years now, supplying the main e-learning companies in the Netherlands with custom made solutions.

During the years we developed a model-driven framework that allows us to deliver stable solutions and at the same time maintain great flexibility in structure and design. The solution is well perceived by the market. Our partners praise the way they can use their high-end designs, have shorter development time and, ofcourse, the interactive elements of this platform.

We're on the verge of launching an online authoring tool and learning management system, that is built on the same core.

5 years ago we made a shift from AS3/PHP to haXe/javascript and haXe/neko.

About the talk

In our talk we'll discuss our model driven framework. And how we use haXe to build it. Below are listed the subjects.

Invariance versus variance

From the experience of developing projects we developed our view on what is the same over all the projects and what differs. On what is the same we'll standardise and on what differs we make flexible.

Model driven

We found that the best way to implement invariance in projects is to enable a model driven framework.

Application flow

The model is not just used for a mapping to a relational database, in the model there is also information on how the user flows through the application, allowing us to map client side views to the objects in the model.

One backend

A result of the model driven approach is a single backend will serve all the projects. So no custom code needed for object relation management.

One language

The backend and the client code is written in haXe, resulting in flexible resourcing. Developers can work on backend and frontend.

We use Neko in the backend and haxeJs in the frontend.

Core

The target for the client applications is javascript and for the server applications it is neko. They are both build on a large target independent core. This gives us a feeling of security for a future "HTML5 is dead" outbreak.

No code generation

We represent the model of the project at runtime in the backend and frontend, this way we do not need to generate code from the model and we can use 1 backend to server multiple projects.

Standalone applications

Within the framework it is possible to make standalone applications that run detached from a database and possible a webserver.