

0.1 Motivation

Die zentrale Modellierung von Domainmodellen ist sehr verbreitet in der Entwicklung von großen Software-Projekten und zentraler Bestandteil von Product Line Engineering. Dabei stellt die Modellierung des Domainmodells einen Kompromiss zwischen der kompletten Modellierung einer Software und der klassischen Entwicklung ohne Modelle dar.

Ziel dieses Kompromisses ist die Effizienz und Sicherheit der Codegenerierung für das Datenmodell einzusetzen, um die Entwicklung der restlichen Software zu vereinfachen.

0.1.1 Product Line Engineering (PLE)

Product Line Engineering befasst sich mit der Entwicklung von mehreren verwandten Softwareprodukten. Dabei handelt es sich häufig um Software für Teilaufgaben und angepasste Kundenversionen der Standardsoftware.

Hierbei besteht für eine Organisation die Gefahr, viele Komponenten mehrfach zu entwickeln und zu verwalten. Durch gemeinsam genutzte Komponenten (Assets) wird die Entwicklung vereinfacht und die Software verhält sich beim Kunden einheitlich. Datenmodelle stellen im **Product Line Engineering** wichtige Assets dar. Einheitliche Modelle verhindern das Übersetzen zwischen verschiedenen Produkte.

0.1.2 EMF

EMF(Eclipse Modelling Framework) ist ein häufig Framework zur Modellierung von Modellen in Java. Es lassen sich große Modelle darstellen und mithilfe von Maven Workflows können diese durch das Build Tool übersetzt werden.

EMF bietet dabei jedoch keine Wahl bei der IDE oder der Programmiersprache. Dies führt dazu, dass Projekte und ganze Firmen bei ihren bisherigen Technologien stehen bleiben. Es wird bei Neuentwicklungen nicht mehr die gefragt, was wären die besten Technologien um das Problem zu lösen, sondern es wird gefragt, wie lösen wir das mit unserer bisherigen Architektur.

0.1.3 Effekte eines unflexiblen Frameworks

Dieser fehlerhafte Ansatz schädigt das Projekt auf mehreren Ebenen:

1. Wissen und Erfahrung Konzentrierung

Da nur eine Architektur in Betracht gezogen wird, hat jedes Mitglied des Teams nur Erfahrung mit der aktuellen Architektur und jegliche Erfahrung mit anderen Technologien verfällt mit der Zeit. Dies schränkt die die Perspektiven auf Probleme sehr stark ein und macht einen Wechsel sehr aufwendig.

2. Schrinkender Bewerberzahl

Da nur Bewerber für die gewählten Technologien in Betracht gezogen werden, verringert sich die Anzahl stark. Der Effekt wird verstärkt, wenn die Technologien als veraltet gelten. Eine kleinere Bewerberanzahl zwingt Unternehmen auch Bewerber, die andernfalls nicht beachtet worden wären, in Betracht zu ziehen. Dies führt zu weiteren Negativen Effekten, da einige schlechte Angestellte die Produktivität vieler guter Angestellter stark senken können.

3. Anfälligkeit gegenüber Sicherheitslücken

Eine starke Festlegung auf Technologien führt dazu, dass Sicherheitslücken gleich jedes Projekt betreffen. So könnten bei einer Zero-Day-Lücke direkt mehrere Schicht im SSchweizer Käse Modell"(Source suchen) wegfallen. Diese Anfälligkeit wird stark erhöht sobald eine Technologie nicht mehr aktiv weiterentwickelt wird. Dies führt häufig dazu, dass andere Updates auch nicht genutzt werden können.

0.2 Aufgabenstellung

Das Domain Modell Framework(DMF) soll es ermöglichen Datenmodelle zentral zu modellieren, sodass diese von verschiedenen Software Projekten genutzt werden können. Dabei soll die Flexibilität besonders beachtet werden, um die bisher bestehenden Nachteile zu vermeiden. Zur Flexibilität gehört die (möglichst) freie Wahl der Programmiersprache und die freie Wahl der Entwicklungsumgebung.

Ziel ist es das DMF für Java und Typescript zu implementieren. Es sollen primär IntelliJ und Visual Studio Code unterstützt werden.