

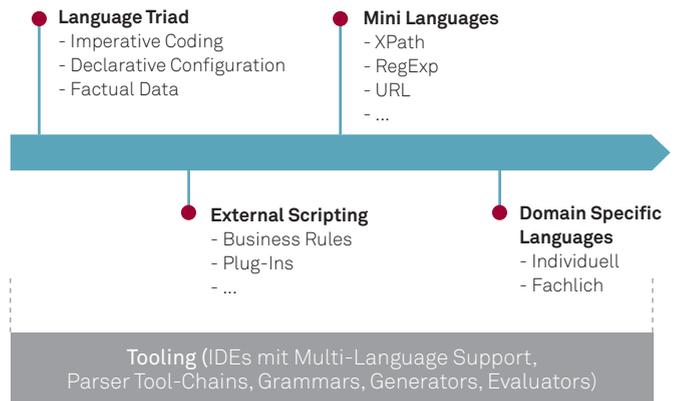
Polyglot Programming

Für jeden Einsatzzweck und jeden Experten die geeignete Sprache

Wie beim Bau eines Hauses sind bei der Entwicklung einer Software unterschiedliche Experten beteiligt, die unterschiedliche Komponenten erstellen. Werden bei der Entwicklung einer Software dementsprechend verschiedene Programmiersprachen je Komponente eingesetzt, spricht man von Polyglot Programming.

Definition

Polyglot Programming ist ein Ansatz in der Software-Entwicklung, der sich dadurch auszeichnet, dass mehrere Programmiersprachen für die Entwicklung einer einzelnen Anwendung verwendet werden. Dieser Ansatz ermöglicht in doppelter Hinsicht die Nutzung der am besten geeigneten Sprache für jeden Zweck – bezogen auf die zu erstellende Funktionalität und bezogen auf den Entwickler. So lassen sich je nach Nutzungsumfang mehrere Vorteile realisieren.



Verwandte Trends

- „Polyglot“ (z.B. Polyglot Persistence)
- Model-Driven Development
- Parser Tool-Chains

Motivation

- „Best tool for the job“
- „Best tool for the developer“
- Separation of Concerns
- Exakter und knapper Code

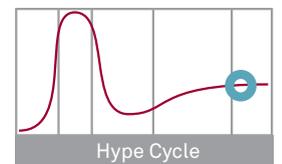
Polyglot Programming

In der kleinsten Ausbaustufe werden Vertreter der verschiedenen Sprachklassen für eine kompakte und abstrakte Beschreibung und Umsetzung des Architekturprinzips „Separation of Concerns“ eingesetzt. Imperative Sprachen eignen sich für Beschreibungen eines Verhaltens (wie), deklarative Sprachen hingegen für Beschreibungen von Zuständen (was). Mit Konfigurationssprachen werden „Hartkodierungen“ vermieden. Für die Oberflächen, Geschäftslogik und Datenpersistenzen setzen die jeweiligen Entwickler verschiedene Sprachen ein. Diese Form des Polyglot Programming ist allgemeine Praxis und wird häufig durch Laufzeitumgebungen wie Web-Browser und Application Server oder durch Frameworks vorgegeben.

In der nächsten Stufe wird das Verhalten einer Anwendung nicht nur konfigurierbar, sondern auch erweiterbar und anpassungsfähig, indem eine imperative Sprache eingesetzt wird. Dadurch können nachträglich beispielsweise Geschäftsregeln durch „Scripting“ beeinflusst werden.

Bei einem weiteren Nutzungsgrad werden Konstrukte aus Mini-Languages, wie Regular Expressions und XPath-Angaben, innerhalb einer Programmiersprache eingebettet, um oft nur eine einzige Aussage in der optimalen Form kompakt auszudrücken.

In der umfangreichsten Form wird die Einbindung des relevanten Fachbereiches in die Entwicklung durch Domain-Specific Languages (DSLs) ermöglicht. Die Fachexperten können mit vertrauten und technisch einfachen Konstrukten selbständig die fachlichen Aspekte, wie Regel- und Testfälle, beschreiben und diese direkt als fertige Teile der Anwendung beisteuern.



Reifegrad

Die Werkzeuge für die Entwicklung, sowohl von DSLs als auch mit (Mini-)Sprachen, haben einen hohen technischen Reifegrad erlangt. Dennoch wird häufig eine Kombination von mehreren Sprachen vermieden, weil einzelne Programmierer nicht mehrere Sprachen gleich gut beherrschen, die Vorteile nicht erkannt werden oder um eine subjektiv wahrgenommene Komplexität zu verhindern.

Marktübersicht



Für Polyglot Programming können Vertreter aller Sprachfamilien wie beispielsweise der imperativen und der deklarativen Programmierung verwendet werden. Erweiterbare IDEs, wie Eclipse, ermöglichen die Kombination von Editoren für mehrere Sprachen.

Alternativen

Eine gewisse Vielfalt von Programmiersprachen ist durch die verschiedenen Laufzeitumgebungen für die Komponenten einer Anwendung vorgegeben. Eine direkte Programmierung in mehreren Sprachen kann vermieden werden, indem eine generische Sprache oder ein graphisches Modellierungswerkzeug für die Beschreibung der Anwendung verwendet werden, aus denen der Quellcode in den jeweiligen Sprachen generiert wird. Dadurch ist der Entwickler jedoch in der Regel in der Vielfalt der Möglichkeiten und in der Optimierung des Codes eingeschränkt.

Beim Verzicht auf eingebettete Mini-Sprachen muss die Funktionalität in der Standardsprache umgesetzt werden. Dadurch wird der Quellcode umfangreicher, häufig fehleranfälliger, weniger kompakt und schwerer verständlich.

msg systems ag

Robert-Bürkle-Straße 1 | 85737 Ismaning/München
 Telefon: +49 89 96101-0 | Fax: +49 89 96101-1113
www.msg-systems.com | info@msg-systems.com

Eine Übernahme von Programmieraufgaben durch den Fachbereich ohne Einsatz von DSLs ist nur dann möglich, wenn entweder die Fach-Experten die verwendete allgemeine Programmiersprache beherrschen oder die Entwickler den Input des Fachbereiches in die Programmiersprache übersetzen.

Referenzszenario

Ein typisches Beispiel für Polyglot Programming ist die Entwicklung von dynamischen Web-Anwendungen. Für das Frontend werden HTML, CSS und JavaScript eingesetzt, während das Backend mit Java, XML, SQL, HTTP und JSON umgesetzt wird. Innerhalb des Java-Codes erfolgen SQL-Abfragen gegen eine relationale Datenbank beziehungsweise Auswertungen von XML-Dokumenten mittels XPath-Angaben. Die Geschäftslogik kann durch den Fachbereich mittels einer DSL beeinflusst und in die Java-Anwendung integriert werden.

Business Impact

Da der Fachbereich Anwendungen selbständig kontinuierlich anpassen kann, wird eine höhere Agilität der Unternehmensprozesse ermöglicht. Zusätzlich tragen die Konfigurierbarkeit und bessere Wartbarkeit zu Kosteneinsparungen bei.

Pro	Contra
Für jeden Zweck wird die am besten geeignete Sprache verwendet.	Der Auswahlprozess der Sprachen erhöht den Projekt-Aufwand.
Durch externe Konfigurierbarkeit ergeben sich Erweiterbarkeit und Anpassungsfähigkeit.	Eine große Vielfalt an Sprachenkenntnissen wird bei der Entwicklung und Wartung benötigt.
Mittels DSLs ist die Einbindung des Fachbereiches möglich.	Unterstützung für mehrere Sprachen in der Entwicklungsumgebung ist erforderlich.
Quellcode ist knapper, exakter, leichter verständlich, häufig weniger fehleranfällig.	Das Debugging wird deutlich erschwert.

