

Open–Source als Basis für den Vergleich von Softwarearchitektur-Beschreibungssprachen

Jan Weinschenker¹, Olaf Zukunft²

¹ Resco GmbH, Millerntorplatz 1,
20359 Hamburg jan.weinschenker@resco.de

² HAW Hamburg, Department Informatik, Berliner Tor 7,
20099 Hamburg, zukunft@informatik.haw-hamburg.de

Abstract: Die explizite Beschreibung von Softwarearchitekturen benötigt eine adäquate Notation. Derzeit existieren verschiedene Vorschläge für Beschreibungssprachen, deren Vergleich bisher auf Basis von konzeptuellen Merkmalen erfolgte. In diesem Beitrag wird ein neuer anwendungszentrierter Ansatz zum Vergleich von Softwarearchitektur-Beschreibungssprachen präsentiert, der auf dem Open–Source Informationssystem PetStore als Modellierungsgegenstand beruht.

1 Einleitung

Die explizite Beschreibung von Softwarearchitekturen hat nach ersten Forschungsarbeiten spätestens mit der Veröffentlichung von [SG96] auch Eingang in Lehre und industrielle Praxis gefunden. Aktuelle Forschungsansätze [Has06] beschäftigen sich u.a. mit der Frage der adäquaten Notation von Softwarearchitekturen. Dafür sind verschiedene „Architecture Description Languages“ (ADL) entwickelt worden, die die unterschiedlichen Aspekte einer Softwarearchitektur beschreiben können. Prominente Vertreter von ADL sind u.a. Armani [Mon99], π –ADL [MO06] und xADL [KGO01]. Aber auch UML [OMG06] wird gelegentlich als ADL verwendet und bietet sich aufgrund der hohen Verbreitung als Referenz für einen Vergleich mit dedizierten ADL an. Generell erlauben alle ADL eine Beschreibung der Struktur von Softwaresystemen. Typische Modellierungspri-mitive sind dabei Komponenten und Konnektoren. Häufig ermöglichen die ADL auch Verhaltensbeschreibungen, die wie bei π –ADL mehr oder wie bei UML weniger formalisiert sind.

Ein potentieller Nutzer solcher ADL benötigt Informationen, um eine für seine Zwecke verwendbare ADL auszuwählen. Ein existierender Vergleich ([MT00]) hat sich insbesondere auf rein konzeptuelle Kriterien konzentriert. Ein Vergleich auf Basis der Beschreibung eines repräsentativen Anwendungssystems liegt bisher nicht vor und soll hier erstmals präsentiert werden.

2 Java™ PetStore

Der Java™ PetStore¹ ist eine Open–Source Referenzapplikation von Sun Microsystems. Fachlich realisiert er eine typische E-Commerce Anwendung, in der Haustiere über das Web verkauft werden können. Technisch sollen mit dem PetStore die Möglichkeiten und Funktionen einer nach Java EE-Architektur entwickelten Anwendung veranschaulicht werden. Da mit dem PetStore diverse Architekturmuster demonstriert werden sollen, sind in dieser Applikation Softwarearchitekturkonzepte erkennbar umgesetzt. Es wurde nicht primär auf Kriterien wie hohe Performanz oder geringer Speicherbedarf geachtet, sodass die Struktur nicht durch Optimierungen verwässert wurde.

Es kann davon ausgegangen werden, dass es sich beim PetStore um eine allgemein bekannte und verbreitete Anwendung handelt. Dafür spricht auch, dass bereits mehrere Varianten des PetStore u.a. von Microsoft existieren. Diese sind zum Teil in anderen Programmiersprachen, wie Microsoft C# für die .NET-Architektur verfasst oder für spezielle Open–Source Applikations-Server wie JBOSS entworfen worden. Auch hier war die Motivation, den PetStore als Hilfsmittel für den Vergleich von Programmiersprachen oder Applikationsservern zu nutzen. Der PetStore ist aufgrund dieser großen Bekanntheit und Verbreitung ein gut geeigneter Testgegenstand im Rahmen dieser Arbeit.

Stark vereinfacht setzt sich der PetStore aus zwei Hauptkomponenten zusammen:

1. Einem Web-Frontend, nachfolgend als PetStoreWebsite bezeichnet und einem
2. Backend, nachfolgend als FulfillmentCenter bezeichnet.

Die PetStoreWebsite richtet sich konzeptionell eher an den Kunden des PetStore, während über das FulfillmentCenter die internen Abläufe des PetStore abgewickelt werden. Die Benutzungsoberflächen sind webbasiert. Die einzelnen Komponenten kommunizieren untereinander über Java-JMS, also über den Austausch asynchroner Nachrichten, die in XML kodiert sind. Die Kommunikation mit dem Kunden erfolgt über E-Mail.

Unabhängig von diesen beiden Komponenten ist der PetStore größtenteils nach der MVC-Architektur gestaltet. Controller und View erstrecken sich sowohl auf PetStoreWebsite und FulfillmentCenter. Das Modell ist im FulfillmentCenter implementiert und wird von dort aus durch die PetStoreWebsite verwendet. Ein Großteil der Geschäftslogik ist in der OrderFulfillment-Komponente implementiert. Diese unterteilt sich laut [SSJ02, Kap. 11] in folgende drei Unterkomponenten:

- Order Process Coordinator (OPC)
- Administrator
- Supplier

Eine ausführliche Beschreibung findet sich in [SSJ02, Kap. 11]. Unter [SUN06] findet sich neben der Dokumentation ebenfalls der Quellcode der Anwendung.

¹In diesem Aufsatz wird die Version 1.3.2 des PetStore verwendet.

3 Vergleich

Zum Vergleich der ADL wurde ein Vorgehensmodell entwickelt, das auf der Modellierung der dargestellten stabilen Version des PetStore mit Hilfe der jeweiligen ADL basiert und die entwickelten Modelle auf Basis einer erweiterbaren Menge von Kriterien bewertet. Zunächst werden die vier Modelle des PetStore in den jeweiligen ADL vorgestellt. Anschließend wird das exemplarische Vorgehen für ein Kriterium beschrieben.

3.1 Modelle

Die vier ADL verfolgen unterschiedliche Ziele. UML ist tatsächlich keine dedizierte ADL, kann aber mit Hilfe von Stereotypen wie in Abbildung 1 dargestellt zur Architekturnotation verwendet werden. Armani ist insbesondere als Austauschsprache zwischen ADL konzipiert und versucht dazu, alle Sprachelemente von ADL zu integrieren. xADL betont insbesondere Erweiterbarkeit als Qualitätsmerkmal und basiert auf einer XML-Darstellung statt einer grafischen Repräsentation. π -ADL ist ein explizit auf die Darstellung von Struktur und Verhalten einer Architektur ausgerichtetes Element einer größeren Sprachfamilie und wird hier in Abbildung 2 exemplarisch dargestellt. Die entwickelten Modelle sind hierarchisch aufgebaut, d.h. komplexe Komponenten enthalten ihrerseits eigene Architekturstrukturen.

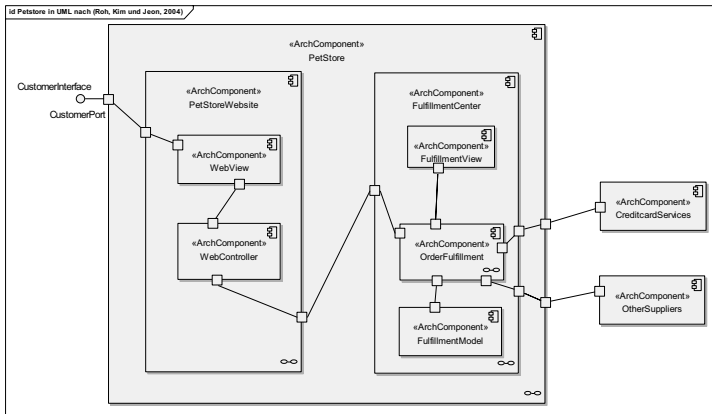


Abbildung 1: PetStore in UML

3.2 Bewertungskriterien

Die exemplarische Modellierung des PetStore mit verschiedenen ADL ermöglicht einem Interessierten bereits, einen ersten Eindruck von den Möglichkeiten der ADL zu erhalten.

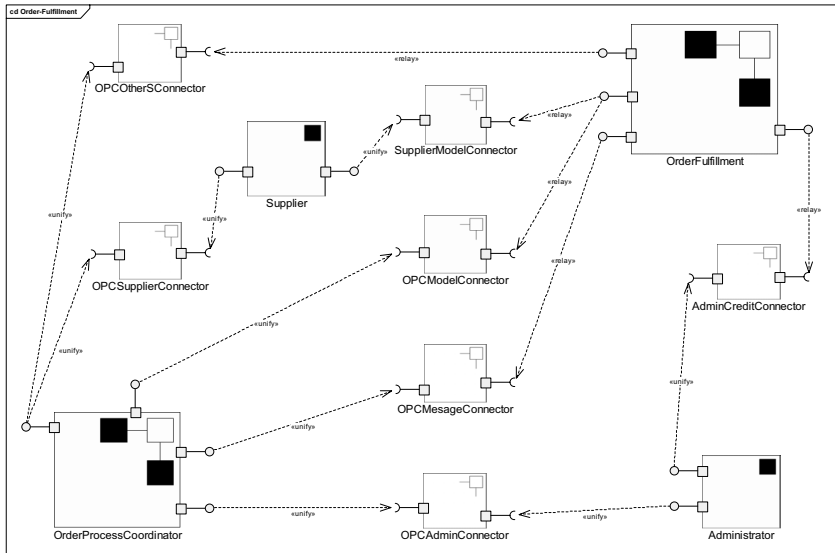


Abbildung 2: PetStore in π -ADL

Darüber hinaus haben die Nutzer von ADL jedoch zumeist weitere Anforderungen, die von Projekt zu Projekt unterschiedlich sein können. Zu diesen Anforderungen gehören Merkmale wie Toolunterstützung, Visualisierung, Weiterentwicklung und Nutzbarkeit. Dazu wurden Bewertungskriterien entwickelt, die zudem noch die Zielsetzung der ADL (z.B. Generierung, Dokumentation, Analyse) sowie Syntax und Semantikbeschreibung der ADL kategorisieren. Aus Platzgründen kann hier nur ein Kriterium skizziert werden. Das in Ta-

Kriterium	Bewertungsskala
Visualisierung vorhanden	ja / nein
Visualisierung UML-konform möglich?	Profil / Metamodell / nein
Vollständige Darstellung aller Elemente der ADL	ja / nein
Strukturelle Darstellung	ja / nein
Darstellung hierarchischer Beziehungen	ja / nein
Darstellung von Verhalten	ja / nein
Darstellung von Typen und Instanzen	ja / nein

Tabelle 1: Bewertungskriterien für die Visualisierung von ADL

belle 1 dargestellte Kriterium der Visualisierung ist insbesondere dann wichtig, wenn die ADL nicht als Austauschformat verstanden werden will, sondern unmittelbarer Ergebnistyp eines Softwarearchitekten ist. Für weitere Details des entwickelten Vorgehensmodells sei auf [Wei06] verwiesen.

4 Bewertung und Ausblick

Die Nutzung des Open-Source Informationssystems PetStore hat sich als optimales Werkzeug zum Vergleich von ADL erwiesen. Es ist dem Open-Source Charakter eigen, dass ADL mit dem Ziel der Generierung von Source-Code hierdurch ideal eingesetzt werden können, weil die Zielstruktur des Generats klar ist. Ein Closed-Source System ist hierfür nicht einsetzbar. Dazu kommt, dass der PetStore nicht nur im Java-Bereich bekannt ist, sondern durch seine Open-Source Eigenschaft auch für .NET-Entwickler verfügbar ist und unter Softwarearchitekten einen hohen Bekanntheitsgrad aufweist. Der PetStore konnte mit allen untersuchten ADL modelliert werden. Auch konnten die jeweiligen Stärken der ADL wie hoher Visualisierungsgrad bei Armani oder formale Semantik bei π -ADL in der Modellierung jeweils genutzt werden. Als nächstes soll untersucht werden, welche weiteren Bewertungskriterien für industrielle Anwender von ADL relevant sind. Auch sollen weitere ADL in die Evaluierung mit aufgenommen werden.

Literatur

- [Has06] W. Hasselbring: Software-Architektur. In: *Informatik Spektrum* 29 (2006), Februar, Nr. 1, S. 48–52.
- [KGO01] Khare, R.; Guntersdorfer, M.; Oreizy, P; Medvidovic N. und Taylor, R.: xADL: Enabling Architecture-Centric Tool Integration with XML. In: *HICSS '01: Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)-Volume 9*. Washington, DC, USA : IEEE Computer Society, January 2001, S. 9053.
- [MO06] Mateescu, R. und Oquendo, F.: π -AAL: An Architecture Analysis Language for Formally Specifying and Verifying Structural and Behavioural Properties of Software Architectures. In: *SIGSOFT Softw. Eng. Notes* 31 (2006), Nr. 2, S. 1–19.
- [MT00] Medvidovic, N. und Taylor, R.: A Classification and Comparison Framework for Software Architecture Description Languages. In: *IEEE Transactions on Software Engineering* 26 (2000), Nr. 1, S. 70–93.
- [Mon99] Monroe, R.: *Rapid Development of Custom Software Architecture Design Environments*. Pittsburg, PA, School of Computer Science, Computer Science Department, Carnegie Mellon University, Dissertation, August 1999
- [OMG06] Object Management Group January 2006. – URL <http://www.omg.org/docs/html/06-01-01/Output/06-01-01.htm>.
- [SG96] Shaw, M. und Garlin, D.: *Software Architecture. Perspective on an Emerging Discipline* Prentice-Hall 1996
- [SSJ02] Singh, I.; Stearns, B. und Johnson, M.: *Designing enterprise applications with the J2EE platform*. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 2002.
- [SUN06] Sun Microsystems, Inc: *blueprints: Java Pet Store Reference Application*. 1. Mai 2006. – URL <https://blueprints.dev.java.net/petstore/>
- [Wei06] Weinschenker, J.: *Beschreibungssprachen für Softwarearchitekturen*. Masterarbeit Department Informatik der HAW Hamburg 2006