

## Abstract

---

# Referenzapplikation für .NET Enterprise Applications

### Name der/des Studierenden

Krummenacher, Roland

### Name der/des Betreuer/in

Huser, Hansjörg

### Name des externen Partners

-

### Master Research Unit und Fachgebiet

Software and Systems, ICT

### Semester

Herbstsemester 2009/2010

### Abstract der Projektarbeit

Patterns und Best Practices, also Methodiken, wie man bekannte, in der Software-Entwicklung immer wiederkehrende Probleme löst, sind aus dem Alltag eines Programmierers nicht mehr wegzudenken. Viele dieser Design- und Architektur-Patterns werden oft unbewusst angewendet, weil sich die modernen Entwicklungsumgebungen an solchen bewährten Mustern ausrichten. So auch das .NET-Framework, welches bereits einige bekannte Patterns implementiert hat und die Entwickler animiert diese zu nutzen. Im Certificate of Advanced Studies-Kurs „.NET Enterprise Applications“ des Microsoft Innovation Centers an der HSR wird den Teilnehmern vermittelt, welche bewährten Methodiken existieren, wie diese implementiert werden und wie das .NET-Framework die Entwickler dabei unterstützt. Ziel dieser Arbeit ist es eine Referenzapplikation zu entwickeln, welche die im CAS-Kurs gelehrt Themen praktisch darlegt und aufzeigt, welche Probleme man bei der Implementation der theoretisch gehaltenen Patterns zu lösen hat.

Zu den Anforderungen an diese Referenzapplikation gehört neben der Implementation der Patterns gemäss dem Lehrstoff des Kurses auch die Verwendung der neusten Technologien aus dem .NET-Umfeld. Die Implementation erfolgt daher mit Hilfe der neuen Funktionalitäten aus dem .NET Framework 3.5 wie der Windows Communication Foundation (WCF), der Windows Presentation Foundation (WPF), der Language Integrated Query (LINQ) sowie deren Zusammenspiel mit den Data Access Layern Entity Framework und LINQ2SQL. Auch für die Infrastruktur-Funktionen wie Security, Exception-Handling und Validierung kamen mit den Application Building Blocks der Enterprise Library Komponenten zum Einsatz, die auf bekannten Mustern basieren.

Zu den meist diskutierten Patterns im Bereich Software-Architektur gehört das Data-Transfer-Objects-Pattern, welches es ermöglicht, mit unterschiedlichen, eventuell unbekannt Clients Daten auszutauschen, ohne dass Business-Objekte über die Grenzen der Server-Applikation verteilt werden müssen. Um den Unterschied zwischen einer Applikation mit DTOs und einer ohne DTOs aufzuzeigen, wurde eine zweite Applikation „HsrOrderApp Mini“ entwickelt, die sich an der Architektur der ersten orientiert, aber ohne DTOs auskommt. Diese Applikation beruht auf den „Self Tracking Entities“-Funktionalität des Entity Frameworks 4.0 welches in der Beta-2-Version zur Verfügung steht.

Zu Beginn der Arbeit war nicht klar, ob sämtliche Patterns aus der Theorie in die Praxis umgesetzt werden können, ohne dass an gewissen Schnittpunkten ein Workaround implementiert werden muss. Es zeigte sich, dass die Restriktionen, welche Patterns wie Domain Model oder DTO verlangen, an gewissen Orten kurzfristig eher ein Hindernis sind. Letztlich ist aber eine Architektur entstanden, welche der Theorie in den meisten Punkten gerecht wird.