Um zwischen den Objekten des Domänenpakets und den Objekten des Datenbankpakets zu mappen haben wir einen eigenen „Dynamic Mapper“ geschrieben, der dynamisch zwischen den beiden Schichten vermittelt. Diese Programmkomponente ist also eine Schnittstelle und schafft eine völlige Unabhängigkeit für beide Seiten und die Persistenzschicht kann problemlos ausgetauscht werden.  
  
Die Idee für dieses Vorhaben entstand in einer Besprechung über die Schichtentrennung und das Klassendesign der Software, aber auch unser technische Coach, Herr Wolfgang Auer, unterstützte diese strikte Entkopplung und gab uns entsprechenden Input – etwa bei den Themen Introspektion und Reflexion.  
Besondere Maßnahmen beim Programmieren, etwa die strenge Namenskonvention, mussten über die ganze Entwicklungszeit rigoros eingehalten werden, um eine dynamisches Mapping für alle möglichen Objekte der Model-Schicht zu ermöglichen. Dies ist notwendig, da wir über die Metadaten wie etwa Klassennamen, Methodennamen und Attribute, die Struktur analysieren und dementsprechend weiter vorgehen. Dadurch erfahren wir alle nötigen Informationen, um in beiden Richtungen zwischen Domäne- und Datenbank-Schicht abzubilden. Mapping bedeutet in diesem Fall also, dass wir verwandte Objekte erzeugen. Diese sind teilweise sehr komplex, da sie verschiedene andere Objekte und Kollektionen halten, die in sich wieder Referenzen aufweisen. Kleine Probleme gab es, wenn wir genau durch die gerade genannte Thematik auf das ursprüngliche Objekt gelangen: Es entsteht ein graphentheoretischer Kreis, den wir explizit unterbrechen müssen. Diese Unterbrechung bewerkstelligen wir mithilfe einer HashMap die wir bei jedem rekursiven Aufruf der Funktion mitgeben. In dieser HashMap werden alle gemappten Paare gespeichert und somit kann überprüft werden ob ein Objekt bereits konvertiert wurde.