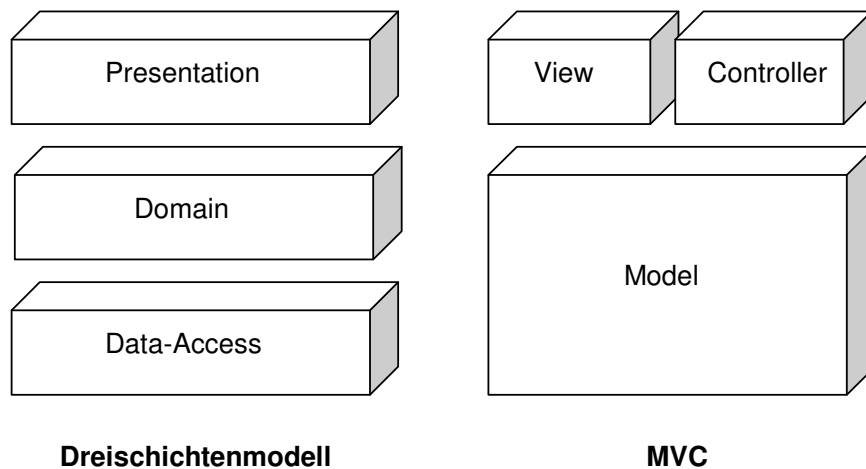


Dreischichtenmodell und MVC

Um eine Software in verschiedenen Einsatzszenarien unverändert betreiben zu können, wird der konstituierende Programmcode gewöhnlich derart entkoppelt, dass man ihn einem der drei Bereiche *Presentation*, *Domain* und *Data-Access* zuordnen kann. Informationen werden zwischen den Bereichen über klar definierte Schnittstellen ausgetauscht, wobei zwischen den Bereichen *Presentation* und *Data-Access* kein direkter Informationsaustausch stattfindet, sondern nur indirekt über den *Domain*. Deshalb spricht man auch von *Software-Schichten* und insgesamt vom *Dreischichtenmodell*.

Das populäre *MVC*-Architekturmuster (*Model*, *View*, *Controller*) übernimmt die Idee der Gruppierung von Programmcode in drei wesentliche Bereiche. Im *Model* sind die beiden Schichten *Domain* und *Data-Access* zusammengefasst, während der *View* und der *Controller* die *Presentation*-Schicht weiter unterteilen.



Das MVC-Architekturmuster eignet sich besonders für die Realisierung von Web-Anwendungen. Bei Desktop-Anwendungen geschieht die Unterteilung der *Presentation*-Schicht in *View* und *Controller* bei manchen eingesetzten Technologien entweder gar nicht oder der Programmierer ist sich dieser nicht bewusst.

Das Dreischichtenmodell betont die in der Praxis wichtige Entkoppelung der *Domain*-Logik von derjenigen Logik, die den Zugriff und die Speicherung von persistent gehaltenen Daten erledigt.

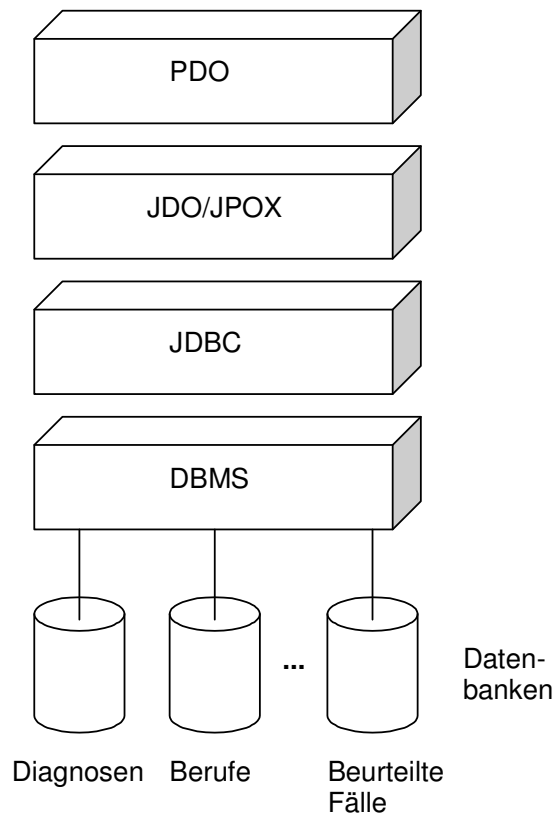
Im folgenden wird dargelegt, wie die TRIAS Underwriting Software dem Dreischichtenmodell und dem MVC-Architekturmuster gehorcht. Dabei gilt zu beachten, dass die Unterteilung der TRIAS Underwriting Software in die beiden Funktionsbereiche *Pflege* und *Anwendung* aus softwaretechnischer Sicht keine Rolle spielt.

TRIAS-Domain

Der *TRIAS-Domain* entspricht im Dreischichtenmodell dem *Domain*. Er enthält einige in der Fachlichkeit der Risikoprüfung begründete Domänen-Objekte und zahlreiche weitere Objekte sowie der mit ihnen verbundenen Programmlogik.

TRIAS-Data-Access

Die *Data-Access* Schicht ist in der TRIAS Underwriting Software wie folgt von oben nach unten in weitere Unterschichten organisiert:



Hierbei bedeuten:

- | | |
|----------|--|
| PDO | Persistent Data Objects, ein von der Triangulum AG entwickeltes Framework, das den Zugriff auf persistente Daten mittels eines ORM (Object-relational mapping)-Tools erlaubt. PDO abstrahiert vom konkreten ORM-Tool soweit dies möglich ist. Deshalb können nach einem einmal zu leistenden Initialaufwand grundsätzlich auch andere ORM-Tools als JDO/JPOX (siehe unten) verwendet werden, also etwa ORM-Tools auf der Basis von JPA (Java Persistence API). |
| JDO/JPOX | JDO steht für Java Data Objects, eine offizielle Sun-Spezifikation für ein herstellerunabhängiges Framework zur persistenten Speicherung von Java-Objekten. JPOX ist die Referenzimplementierung von JDO. |
| JDBC | Java Database Connectivity, eine universelle, standardisierte Schnittstelle (API) zu einer SQL-Datenbank. |
| DBMS | Datenbankmanagementsystem, also etwa Oracle oder DB2, aber auch Anbieter von objektorientierten Datenbankmanagementsystemen. (Unterstützt werden genau diejenigen DBM-Systeme, die von JPOX unterstützt werden.) |

Der konsequente Einsatz von ORM-Tools bedeutet, dass die TRIAS Underwriting Software nicht mit einem konkreten Relationenmodell in Verbindung gebracht werden kann, sondern nur mit einem Objektmodell. Die Abbildung der Objekte in Entitäten, die das Datenbankmanagementsystem versteht (bei relationalen DBM-Systemen sind dies Relationen), ist Sache des ORM-Tools.

TRIAS-Model

Der TRIAS-Domain und die darunter liegende Data-Access-Schicht stellen das Model im MVC-Architekturmuster dar. Physisch besteht das TRIAS-Model, abgesehen vom DBMS und den Datenbanken, aus einer Sammlung von *JAR* (Java Archive) -Dateien.

Das TRIAS-Model verwendet gewöhnliche Java Objekte, sog. POJOs (*Plain Old Java Objects*). POJOs benötigen im Gegensatz zu EJBs (*Enterprise Java Beans*) zur Laufzeit keinen *Application Server*. Dennoch ist das TRIAS-Model auch in einem Application Server lauffähig und kann beispielsweise die Datenbankverbindung über eine auf dem Application Server konfigurierte *JNDI-Datasource* beziehen.

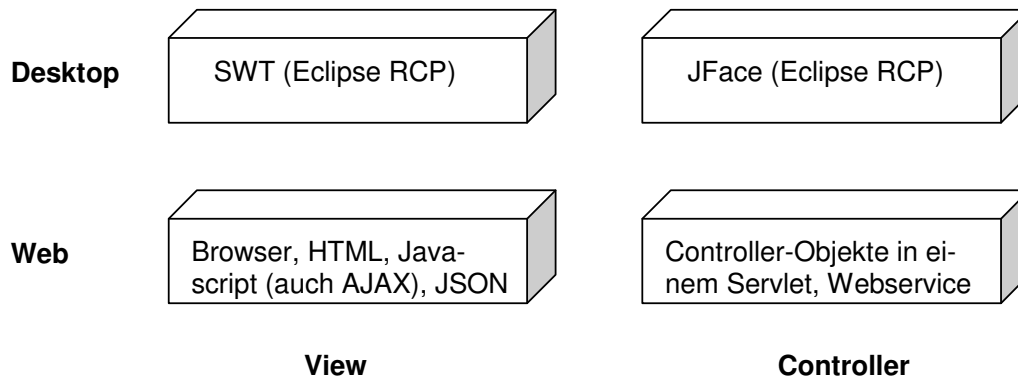
Das TRIAS-Model ist vollkommen unabhängig von der darüberliegenden darstellenden Presentation-Schicht. Dies bedeutet, dass das TRIAS-Model sowohl in einer Desktop-Lösung als auch in einer Web-Anwendung unverändert eingesetzt werden kann. Dadurch nimmt es einen eigenständigen Produktcharakter gemäss der Formel „TRIAS Underwriting Software = TRIAS-Model“ an. Einschränkend muss allerdings gesagt werden, dass die Programmierung der Presentation-Schicht für das TRIAS-Model aufgrund des recht komplexen fachlichen Hintergrunds ein grösserer Aufwand darstellt, siehe dazu das Kapitel „TRIAS-Presentation“.

Der Begriff „TRIAS-Geschäftslogik“ ist synonym zum Begriff „TRIAS-Model“ zu verstehen.

TRIAS-Presentation

Es sei erneut darauf hingewiesen, dass es aus Sicht des TRIAS-Model einerlei ist, welche Technologien auf der Presentation-Schicht zum Einsatz kommen. Da *.Net*-Programmiersprachen und Java mit entsprechenden *Glue*-Technologien zusammengebracht werden können, ist auch der Einsatz einer Programmierumgebung von Microsoft zur Realisierung der Presentation-Schicht möglich.

Im folgenden werden Programmiersprachen, Konzepte und Technologien genannt, mit denen eine Realisierung der Presentation-Schicht der TRIAS Underwriting Software in der Vergangenheit durchgeführt wurde.



Es bedeuten:

- Eclipse RCP Rich Client Platform von Eclipse. Ein Application Framework für den Bau von desktopbasierten Anwendungen, die mittels sogenannter Plugins erweiterbar sind.
- SWT Standard Widget Toolkit. Das von Eclipse gegenüber Swing favorisierte plattformübergreifende *widgit set* für den Bau von grafischen Benutzeroberflächen (GUI).
- JFace Eine in Java geschriebene Sammlung von vorgefertigten, immer wieder einsetzbaren Controller-Objekten.

HTML	Hypertext Markup Language. Die Standardformatierungssprache für Dokumente im Internet.
Javascript	Eine Programmiersprache, die auf dem Client, also im Browser interpretiert wird und deren Programmcode innerhalb von HTML-Seiten platziert ist.
AJAX	<i>Asynchronous JavaScript and XML</i> . Eigentlich eine nicht ganz zutreffende Bezeichnung für eine Zwischendurchkommunikation zwischen Client (Browser) und Server. Der geneigte Leser sei hier auf entsprechende Stellen im Internet verwiesen. Da der Prozess der Eingabe der risikorelevanten Daten sehr benutzerinteraktiv ist, eignet sich AJAX hier ganz besonders. Allerdings ist der Einsatz von AJAX an gewisse technische Voraussetzungen geknüpft.
JSON	Javascript Object Notation. Eine kompakte Formatierungssprache für den Austausch von strukturierten Daten. Da JSON kompakter ist als XML, wird JSON auch alternativ zu XML in einer AJAX-Kommunikation eingesetzt. (Deshalb ist das X in AJAX nicht unbedingt gerechtfertigt.)
Servlet	Programmcode einer Web-Anwendung, der auf dem Server läuft. Die Servlet-Spezifikation ist Teil des Java EE (Java Enterprise Edition) Standards. Servlets werden in der Regel nicht direkt programmiert, sondern mit Hilfe von Web-Application-Frameworks (Struts, JSF, Tapestry, etc.) erstellt. Servlets beinhalten in der Regel neben den Controller-Objekten auch das Model. Für den Betrieb eines Servlets ist ein WEB-Container (Tomcat, Jetty, etc.) oder ein Application Server erforderlich.
Webservice	Gemäss Wikipedia „eine Software-Anwendung, die mit einem Uniform Resource Identifier (URI) eindeutig identifizierbar ist und deren Schnittstellen als XML-Artefakte definiert, beschrieben und gefunden werden können. Ein Webservice unterstützt die direkte Interaktion mit anderen Software-Agenten unter Verwendung XML-basierter Nachrichten durch den Austausch über internetbasierte Protokolle“.

Der Pflgeteil der TRIAS Underwriting Software eignet sich wegen seiner äusserst komplexen Benutzerführung und wegen der potentiell niedrigen Anzahl von Benutzern nicht für das Internet. Sie liegt als Desktop-Applikation auf Basis des Eclipse RCPs vor. (Aus rein technischer Sicht stünde einer webbasierten Datenpflege jedoch nichts im Wege.)

Die Programmierung der webbasierten Anwendung der TRIAS Underwriting Software schlägt mit einem Aufwand von ca. 25 Personentagen zu Buche.