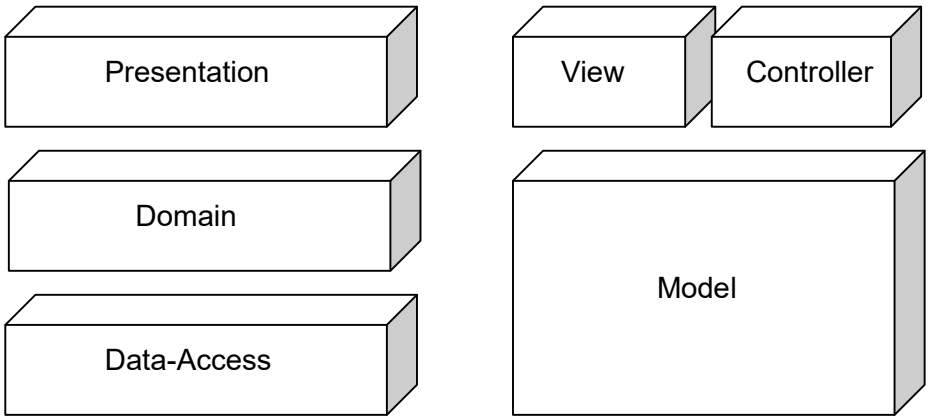
Dreischichtenmodell und MVC

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*.

Auch beim populären *MVC*-Architekturmuster (*Model*, *View*, *Controller*) wird der Programmcode in drei wesentliche Bereiche gegliedert. Im *Model* sind die beiden Schichten Domain und Data-Access zusammengefasst, während der *View* und der *Controller* die Presentation-Schicht weiter unterteilen.



Dreischichtenmodell **MVC**

Das MVC-Architekturmuster und dessen Varianten *MVVM*, *MVP* und *MVA*, auf die hier nicht näher eingegangen wird, eignet sich besonders für die Realisierung von Web-Anwendungen, denn 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 Entkoppelung der Domain-Logik einerseits von derjenigen Logik, die den Zugriff und die Speicherung von persistent gehaltenen Daten erledigt, andererseits von derjenigen Logik, die für die visuelle Benutzerschnittstelle verantwortlich ist. Dies ermöglicht den Einsatz der unterschiedlichsten Technologien – Datenbanktechnologien auf der Data-Access-Schicht, View-Technologien auf der Presentation-Schicht – ohne dass die eigentliche Geschäftslogik, die sich im Domain befindet, tangiert wäre.

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 die dazugehörige Programmlogik.

TRIAS-Data-Access

Die Data-Access Schicht der TRIAS Underwriting Software gehorcht dem *Data Access Object* (DAO) Entwurfsmuster, das gegenüber dem TRIAS-Domain eine klare Schnittstelle in Form von persistierbaren Domain-Objekten und der mit diesen verbundenen Funktionalität (Objekt von der Datenbank lesen, Objekt aktualisieren, Objekt in der Datenbank speichern, etc.) aufbaut. Die Schnittstelle favorisiert keine bestimmte Technologie zur Persistierung der Objekt-Daten. In der Tat liegt die Implementation in dreifacher Ausführung vor: JDO/DataNucleus, Java ACDP, SOI.

JDO/DataNucleus	JDO steht für Java Data Objects, eine Spezifikation für ein herstellerunabhängiges Framework zur persistenten Speicherung von Java-Objekten. DataNucleus ist die Referenzimplementation von JDO, die auch das ORM Konzept (Object-Relational Mapping) unterstützt. Mit JDO/DataNucleus können die persistenten Daten der TRIAS Underwriting Software in einer relationalen Datenbank (bspw. Oracle, MySql) gehalten werden.
Java ACDP	ACDP steht für Application Controlled Data Persistence und bezeichnet eine bestimmte, nicht SQL basierte Vorgehensweise zur Persistierung von Objektdaten. Mit der verwendeten Java Implementation können Objektdaten besonders effizient geladen, gespeichert und aktualisiert werden, da auch die TRIAS Underwriting Software in Java programmiert ist. Zudem eignet sich Java ACDP für den nur lesenden, in die Anwendung eingebetteten Betrieb. Details siehe unter https://acdphome.github.io/.
SOI	SOI steht für Secure Open Interface und ist eine von der Triangulum AG eigens für die TRIAS Underwriting Software entwickelte, auf JSON (Javascript Object Notation) basierende Datenbank für den nur lesenden, in die Anwendung eingebetteten Betrieb. SOI wird möglicherweise in der Zukunft durch Java ACDP abgelöst werden.

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 möglicherweise eingesetzten Datenbankmanagementsystem (DBMS) und den Datendateien, aus einer Sammlung von *JAR* (Java Archive) -Dateien.

Das TRIAS-Model ist vollkommen unabhängig von der darüber liegenden 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.

Zum Betrieb des TRIAS-Model als Web-Anwendung oder Web-Service ist ein einfacher *Servlet Container* (bspw. Tomcat, Jetty) ausreichend. Ein Application Server (bspw. WildFly Application Server, IBM WebSphere) ist für den Betrieb nicht notwendig.

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 (oder des Domain im Dreischichtenmodell) einerlei ist, welche Technologien auf der Presentation-Schicht oder im View/Controller zum Einsatz kommen.

Im Folgenden werden Programmiersprachen, Frameworks, Libraries, Plattformen, Konzepte und Technologien aufgelistet, mit denen die TRIAS-Presentation in der Vergangenheit umgesetzt wurde. Für sämtliche Details sei auf die Wikipedia verwiesen.

Desktop-Anwendung: SWT, JFace (Eclipse RCP), JavaFX (nicht mehr unterstützt)

Web-Anwendung: HTML, CSS, Javascript, AJAX, JSON, React, Angular, Typescript, Bootstrap, W3.CSS, handlebars, Mustache, JQuery

Der Pflegeteil der TRIAS Underwriting Software ist gegenwärtig nur als Desktop-Anwendung auf Basis des Eclipse RCPs umgesetzt. Zurzeit wird aber an einer webbasierten Lösung gearbeitet, um die Pflege der Wissensbasis auch in der Cloud zu ermöglichen. Der Pflegeteil der TRIAS Underwriting Software wird naturgemäss nur von sehr wenigen Anwendern gleichzeitig benutzt.

Was den Funktionsbereich der reinen Anwendung der TRIAS Underwriting Software betrifft, so stehen für die Umsetzung des View/Controllers drei Varianten von Schnittstellen zur Verfügung, die im Webauftritt im Kapitel „Technik/Integration“ beschrieben sind.