

Praxisprojekt: t9203_inventor_interface

Formale Daten

Name: Moritz Schäfer

Matrikelnummer: 11152026

Email: moritz.schaefer@smail.th-koeln.de

Zeitraum: Vom 01.10.2024 bis zum 28.02.2025 mit 16 Stunden pro Woche

Firmenanschrift: Friedrich Lütze GmbH, Ettore-Bugatti-Straße 29, 51149 Köln

Ansprechpartner: Tobias Saalfeld, tobias.saalfeld@luetze.de, +49 7151 6053-748

Projekteinleitung

Das Projekt soll die Möglichkeiten zur Automatisierung der Autodesk-Anwendung Inventor und Vault für die Systeme der Firma Lütze im Rahmen meiner studentischen Tätigkeit prüfen. Lütze benutzt Inventor zur Entwicklung von Bauplänen für die Fertigung. Ein Inventor-Projekt setzt sich grob aus drei grundlegenden Dateitypen zusammen. Die Part-Datei (.ipt) für ein einzelnes Bauteil. Die Assembly-Datei (.iam), für das gesamte Baustück mit allen Bauteilen und die Drawing-Datei (.idw, .dwg), der letztliche Bauplan mit Profil, Draufsicht und Sprengzeichnung. Im Bauplan sind Informationen tabellarisch im Zeichnungsrahmen dargestellt. Wenn Lütze die Zeichnung an Kunden rausgeben möchte, müssen kritische Informationen händisch entfernt werden. Dies soll in Zukunft automatisiert geschehen. Zunächst soll ich eine Information aus einem Rahmen auslesen. Dies ist bereits mit einem Python-Skript gelungen. Herausfordernd war dabei die Einarbeitung in die Aufrufstruktur der Objekte der Anwendung. Die Pythonerweiterung PyWin32 konnte dabei auf die VBA-API von Inventor zugreifen. Der ApprenticeServer ist eine ReadOnly-Instanz von Inventor, die einen performanten Zugriff auf die API ermöglicht.

Nun soll ich das Datenverwaltungstool Vault mit einem Skript ansprechen. Die zentrale Komponente von Vault ist der ADMS, dessen Abhängigkeiten in der folgenden Skizze dargestellt sind. Die Zeichnungsdaten sind bei Lütze zentralisiert über Vault erreichbar. Eine geeignete Schnittstelle hierfür ist die Webserver-Komponente der Anwendung. Vault bietet wie Inventor Dynamic-Link-Libraries (DLL) zum Verbindungsaufbau zu den jeweiligen Komponenten. Die DLLs sind mit einem Skript von C# oder PowerShell ansprechbar. Anders als Inventor wird Vault nicht von PyWin32 unterstützt, sodass für den Zugriff auf die API beider Anwendung die Softwareplattform Dotnet (.NET) von Microsoft für die weitere Entwicklung verwendet werden sollte.

Gelingt mir der Zugriff auf die Vault-Schnittstelle, erhalte ich weitere Anweisungen von meinem Betreuer zur Fortführung des Projektes.

Projektskizze

