

Automatische generatie van een TIA Portal project met TIA Portal Openness

Student: Kevin Meersman,

Promotoren: Bert Deslee, Isabel Sweertvaegher

In samenwerking met: ATS

Academiejaar 2022 - 2023

of om data en code te importeren of exporteren als een XML bestand.

I. INLEIDING

A. *ATS*

ATS is een multidisciplinair bedrijf dat diensten levert in elektro & automatisering, mechatronica, HVAC & energietoepassingen en distributie. De business unit (BU) automatisering analyseert, definieert en realiseert projecten in procesautomatisering en motion control. Daarnaast biedt de BU ook ondersteuning aan projecten van andere diensten binnen ATS.

B. *PLC objecten*

Om de ontwikkelingstijd van PLC software in te korten wordt gebruik gemaakt van objecten. Een object bestaat uit code die regelmatig hergebruikt wordt en getest is op correcte werking. Op deze manier kan een deel code geïmplementeerd worden met minimale instellingen en met zekerheid dat de code functioneert zoals verwacht.

C. *TIA Portal Openness*

TIA Portal Openness [1] is een API die toestaat om interactie aan te gaan met TIA Portal aan de hand van een eigen applicatie. Een API is een interface die het mogelijk maakt voor 2 applicaties om met elkaar te communiceren en bestaat uit een reeks van functies gegroepeerd in een bibliotheek. De Openness API laat toe om aanpassingen te maken aan de objecten en instellingen in een TIA project

II. DOELSTELLINGEN

Het doel van dit eindwerk is het ontwikkelen van een TIA Portal Openness applicatie die het opbouwen van een TIA Portal project versnelt door de nodige PLC objecten automatisch te genereren en importeren in het project.

De masterproef kan onderverdeeld worden in enkele delen. In een eerste deel wordt onderzocht hoe de Openness API interactie aangaat met de TIA Portal omgeving en wordt een methode uitgezet om inputdata (lijst van objecten) om te zetten naar objecten. Eens de werkmethode vastgelegd is, zal deze toegepast worden voor het genereren van objecten in de PLC alsook voor het genereren van visualisatieobjecten.

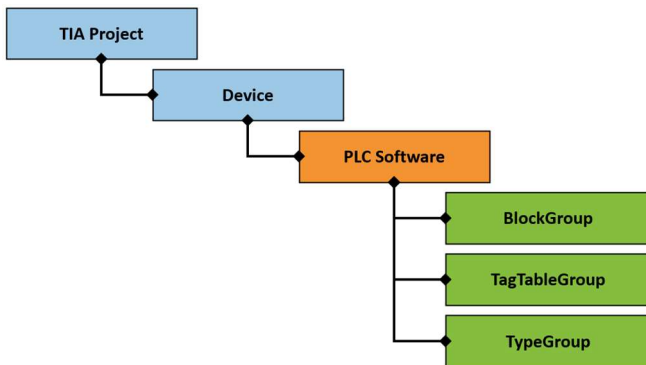
Tijdens het ontwikkelen van de applicatie wordt rekening gehouden met de structuur van de code, zodat een uitbreiding zo vlot mogelijk verloopt.

Nadat er voldoende kennis is opgedaan over de werking van TIA Portal Openness en er een eigen applicatie is uitgewerkt zal TIA Portal Openness vergeleken worden met andere softwaretools van Siemens, die ook de mogelijkheid bieden tot genereren van code en visualisatie.

III. RESULTATEN

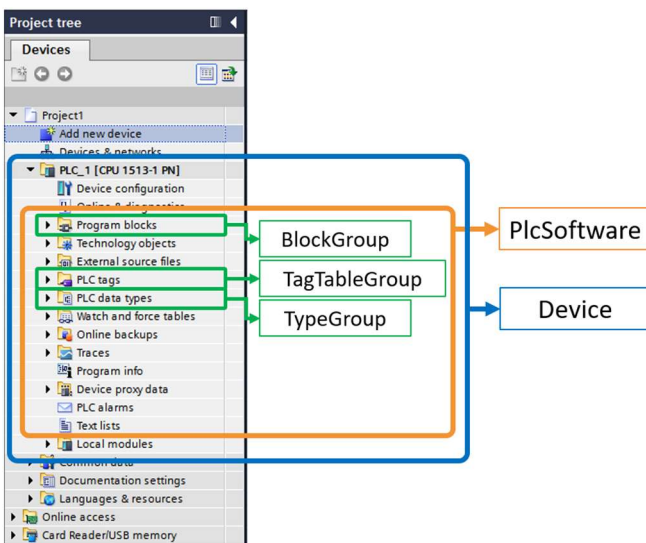
A. TIA Portal Openness

Om aanpassingen te maken aan een TIA Portal project met de Openness API is het belangrijk om het Openness object model te kennen (Figuur 1). Het object model beschrijft een TIA Portal project aan de hand van C# klassen.



Figuur 1: Vereenvoudigde weergave Openness object model

De hiërarchische structurering van deze klassen komt overeen met de structuur terug te vinden in TIA Portal (Figuur 2).



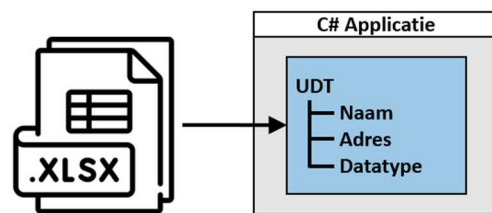
Figuur 2: Overeenkomst TIA Portal en object model

Wanneer het object model gekend is, kan er softwarematig naar het juiste object genavigeerd worden en de bijbehorende functies uitgevoerd worden. Afhankelijk van het type object zullen deze verschillen. Voor het 'Device' object worden de instellingen van het device aangepast. Voor het 'BlockGroup' object kunnen de verschillende types blokken geïmporteerd of geëxporteerd worden of de attributen van de blok aangepast worden.

B. Werkmethode van de applicatie

De te importeren objecten worden aan de applicatie aangeleverd in de vorm van een Excel bestand. Dit bestand dient volgens een vooropgestelde structuur ingevuld te worden.

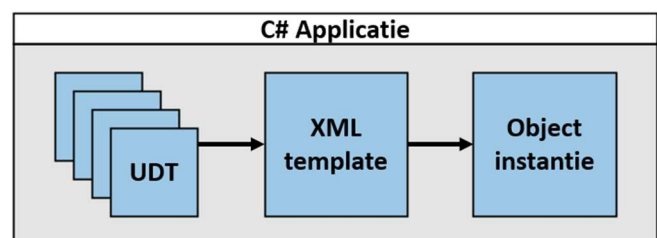
Bij het uitlezen van het Excel bestand wordt elk object opgeslagen in een UDT (User Data Type). Op deze manier wordt alle data gebundeld (Figuur 3). Naast de data uit het Excel bestand bevat de UDT ook nog extra datavelden met informatie die doorheen de applicatie gebruikt wordt.



Figuur 3: Data in Excel naar UDT omzetten

Aanpassingen aan data en interface van een program block (FB, FC, DB) of tag lijst kunnen niet rechtstreeks gemaakt worden via de API. Hiervoor dient een XML bestand [2] opgesteld te worden dat de blok beschrijft.

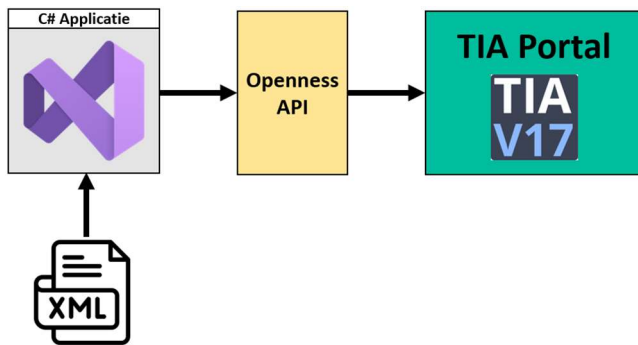
Om de XML code te creëren wordt gebruik gemaakt van zelfgemaakte XML templates. Een template beschrijft één object waarvan de in- en outputs nog niet ingevuld zijn. Het template wordt aangevuld met de data die in de UDT's zijn opgeslagen (Figuur 4). Door alle ingevulde templates bij elkaar te voegen wordt één blok gecreëerd die alle instanties van het object bevat.



Figuur 4: Genereren van objecten met UDT en template

De Openness API staat enkel toe om bestanden van op de PC te importeren. Om deze reden wordt de gegenereerde XML code uitgeschreven naar een '.xml' bestand op de PC.

Als laatste stap worden de '.xml' bestanden geïmporteerd in het TIA Project door de Openness API (Figuur 5). Hier moet opnieuw rekening gehouden worden met het object model. Het '.xml' bestand dat de program blocks bevat moet geïmporteerd worden in de 'BlockGroep'. Het '.xml' bestand van de tag lijsten in de 'TagTableGroep', etc.



Figuur 5: XML bestand importeren naar TIA Portal

C. PLC

De werkmethode beschrijft het stappenplan voor het genereren en importeren van de object-instanties. Maar voor dat deze geïmporteerd kunnen worden, dient de CPU al voorzien te zijn van bepaalde bronbestanden. Zo is het nodig dat de UDT, onder de map 'PLC Data Types', en het object reeds aanwezig zijn in de CPU. Pas wanneer dit het geval is, is de Openness API in staat om de gegenereerde instanties te importeren.

De applicatie kan eveneens nieuwe instanties toevoegen bij reeds bestaande object-instanties. Hiervoor worden de bestaande blokken geëxporteerd als '.xml' bestand. De relevante XML code wordt uit dit bestand gehaald en toegevoegd aan de nieuw gegenereerde XML code.

D. Visualisatie

Bij het importeren van visualisatieobjecten moet de HMI voldoen aan een aantal voorwaarden.

De objecten zijn opgebouwd als 'HMI faceplate'. Deze 'faceplates' dienen in de 'Project Library' te zitten. Bij de CPU werd er doelbewust gekozen om alle bronbestanden op te slaan als XML en niet in een 'Global Library' die dan geopend wordt in het project. Voor de 'Project Library' is er geen manier om te importeren a.d.h.v. XML bestanden. Hier is er geen andere optie dan te werken met een

'Global Library' en deze door de API te laten kopiëren naar de 'Project Library'.

De connectie tussen PLC en HMI moet gemaakt zijn vooraleer er geïmporteerd kan worden. De API bezit echter geen functies om dit te doen. Dit dient manueel te gebeuren door de programmeur of er moet gewerkt worden met een startproject waar de connectie reeds is gemaakt.

Wanneer een scherm wordt geïmporteerd moet de resolutie in het XML bestand overeen komen met de resolutie van de HMI in het project. Het importeren van de PLC objecten is onafhankelijk van het gebruikte type CPU. Bij de visualisatie is dit echter wel het geval. Dit wil zeggen dat er verschillende templates dienen gemaakt te worden voor elke resolutie.

IV. BESLUIT

Uit deze masterproef kan besloten worden dat TIA Portal Openness een krachtige tool is die veel kan bieden bij het ontwikkelen van PLC projecten. Een codegenerator versnelt niet enkel het proces maar elimineert ook een heleboel fouten die gemaakt kunnen worden.

Maar er moet ook vastgesteld worden dat de API zijn beperkingen heeft. Dit wordt vooral opgemerkt bij het genereren van visualisatieobjecten. Bij het genereren van PLC code kan elke stap uitgevoerd worden door de API, van project aanmaken tot code importeren. Bij de visualisatie is er steeds tussenkomst nodig door een programmeur. Een tweede grote beperking is dat de resolutie van het XML bestand overeen moet komen met de resolutie van de HMI in het project.

De complexiteit mag ook niet onderschat worden. Om een applicatie te bouwen op een gestructureerde manier is er een uitgebreide kennis nodig van C# of een andere programmeertaal. Ook de kennis van TIA Portal is belangrijk. In vele gevallen is er niet één persoon die uitgebreide kennis heeft over beide aspecten en daar bovenop ook de link kan leggen tussen de verschillende programma's. Dit heeft als gevolg dat er meerdere personen betrokken moeten zijn bij de ontwikkeling, wat een stijgende ontwikkelingskost met zich meebrengt.

V. REFERENTIES

- [1] *TIA Portal Openness: API for automation of engineering workflows*, 2022.
- [2] AWS, „What is XML? - Extensible Markup Language (XML) Explained - AWS,“ Amazon Web Services, [Online]. Available: <https://aws.amazon.com/what-is/xml/>. [Geopend 24 04 2023].
- [3] P. Smith, „Using XML Templates in TIA Openness | DMC, Inc.,“ DMC, Inc, 18 09 2019. [Online]. Available: <https://www.dmcinfo.com/latest-thinking/blog/id/9930/using-xml-templates-in-tia-openness>. [Geopend 25 04 2023].