

Modular Virtual Commissioning - A learning factory approach focussing on modularity and integration of component models.

Master Thesis

In partial fulfillment of the requirements for the degree

"Master of Science in Engineering"

Master program:
Mechatronics & Smart Technologies
Management Center Innsbruck

Supervisor:
Benjamin Massow, B.Sc. M.Sc.

Author:
Dominique Mathäus Geiger
2010620012

Declaration in Lieu of Oath

„I hereby declare, under oath, that this master thesis has been my independent work and has not been aided with any prohibited means. I declare, to the best of my knowledge and belief, that all passages taken from published and unpublished sources or documents have been reproduced whether as original, slightly changed or in thought, have been mentioned as such at the corresponding places of the thesis, by citation, where the extent of the original quotes is indicated.“

Place, Date

Signature

Abstract

In this master thesis a concept for component libraries is created consisting of CAD data, simulation models, example control code and documentation.

Virtual commissioning is becoming increasingly important in the engineering process of plants and machines. This is the result of time savings and thus reduced costs in product development.

The behavior modeling of the different components can be done in several different tools, mostly depending on the manufacturer. The CAD data can also be available in different file formats and are thus no exception. This variety of possibilities increases the complexity of the integration process enormously. However, in order for virtual commissioning to be as efficient as possible, the integration of the used components should be as simple as possible.

In order to solve this problem, an exemplary library structure is developed in this thesis, which consists of the behavior model, CAD data and code snippets with related documentation. The basis for this are freely available and common interfaces, like Functional Mock-up Interface (FMI) for the behavior model and Collada for kinematic CAD data. This ensures that the widest possible range of applications is achieved. The practical use of this library is further demonstrated in a practical application.

In the example shown, the behavior model is executed in a separate task of the *Beckhoff* PLC, ensuring real-time capability. Visualization is done directly in the CAD software *Autodesk Inventor* based on the current state of the behavioral model. The behavior model is generated in *Matlab Simulink* and integrated using *FMI*.

As shown, using the developed library, the process of virtual commissioning can be simplified without the need for additional expertise.

Keywords: Virtual Commissioning, PLC, Automation, Modularity, Development.

Contents

1. Introduction	1
1.1. Motivation	1
1.2. Aim of this Thesis	1
1.3. Structure of this Thesis	1
2. State of the Art in Virtual Commissioning	2
2.1. Visualization in the PLC Development Environment	2
2.2. Visualization using an Physics Engine	2
2.3. Digital Twin in Simulation Software	2
2.4. Hardware in the Loop (HIL)	2
2.5. Available Data Formats	2
2.5.1. Geometry Data	2
2.5.2. Behavior Models	3
2.5.3. Code Snippets and Examples	3
2.5.4. Documentation	3
2.5.5. Exchange Libraries	3
3. Proposed Data Structure for universal Data Exchange	4
3.1. Structure	4
3.2. Advantages and Disadvantages	4
4. Exemplary Application using the proposed Data Structure	5
4.1. Introduction	5
4.2. Module Separation	5
4.3. Module Filling	5
4.4. Visualization	5
5. Results and Evaluation	6
6. Summary and Outlook	7
6.1. Summary	7
6.2. Outlook	7
Bibliography	III
List of Figures	IV
List of Tables	V
List of Code	VI
List of Acronyms	A1
A. Example Appendix	A1

1. Introduction

1.1. Motivation

Virtual commissioning becomes more and more popular in plant engineering. As a key hindrance the challenging creation of simulation models and their integration in the plant models are identified. Integrating simulation models, kinematized visualizations and according automation programm into a components library would lower the hindrance.

1.2. Aim of this Thesis

Aim of this thesis is to extend an existing learning factory and create a concept where component – e.g. of a conveyor belt and a gantry robot – libraries including simulation models, kinematized visualizations and according automation programm can be integrated easily. The overall plant model should be implemented in unity or a standard CAD system.

Abgrenzung VIBN auf SPS Ebene

1.3. Structure of this Thesis

2. State of the Art in Virtual Commissioning

2.1. Visualization in the PLC Development Environment

The Human-machine interface (HMI) offers a quick possibility for testing the control software. In most cases, a graphical user interface can be created directly in the development environment in which current values and outputs of the control software can be displayed and inputs can be set.

Bei dieser Herangehensweise ahmt der menschliche Entwickler das Verhalten der realen Anlage nach und versucht damit Probleme in der Software zu identifizieren.

Das Erstellen einer HMI wird von vielen gängigen SPS-Herstellern wie zum Beispiel Siemens und Beckhoff unterstützt.

2.2. Visualization using an Physics Engine

Die nächste Ausbaustufe in der virtuellen Inbetriebnahme ist die Verwendung einer externen physics engine wie zum Beispiel Unity. Dabei können die originalen Geometriedaten importiert werden, was eine bessere Übersicht bietet. Vorallem mit vorgefertigten Libraries kann schnell ein Model aufgebaut werden. Nachteilig an dieser Variante ist einen nötige Kommunikation zwischen der PLC runtime und der physics engine.

2.3. Digital Twin in Simulation Software

Die Verwendung einer Simulations-Software wie Matlab/Simulink ist vorallem bei größeren Systemen oder Regelungen vorteilhaft. Vorallem eine graphische Oberfläche bieten eine gute Übersicht und verringern damit den nötigen Zeitaufwand in der Erstellung von Modellen oder ihrer Wartung.

2.4. Hardware in the Loop (HIL)

2.5. Available Data Formats

2.5.1. Geometry Data

Die Industrie und ihre Produkte sind sehr vielfältig und damit auch die Anforderungen an die CAD-Software. Je nach Anwendungsfall kann dadurch eine Software vorteilhafter sein als eine andere. Ein Hauptunterschied dabei ist *direkte Modellierung* im Vergleich zur *parametrisierten Modellierung*. Bei der direkten Modellierung wird die Geometrie mithilfe von fixen Werten erzeugt. Daraus resultieren keine Abhängigkeiten zwischen den

verschiedenen Elementen der Geometrie. Diese Art der CAD-Software wird vor allem im statischen Bereich, wie zum Beispiel im Stahlbau, eingesetzt. Im Gegenzug dazu wird bei der parametrisierten Modellierung die Geometrie mithilfe von Abhängigkeiten und Parametern erzeugt. Dies ist vor allem im dynamischen Bereich mit beweglichen Teilen von Vorteil. Große Anbieter von CAD-Software sind dabei: *Creo Elements*, *Autodesk Inventor* oder *Siemens NX*. Bei dem Import von CAD-Modellen muss auf das Austauschformat geachtet werden, da nicht alle die Kinematisierung über Abhängigkeiten unterstützen. Ein Beispiel, wo die Kinematisierung beibehalten wird, wäre *Collaborative Design Activity (COLLADA)* und ein Gegenbeispiel das *Standard for the Exchange of Product model data (STEP)* oder auch *Initial Graphics Exchange Specification (IGES)* Format.

2.5.2. Behavior Models

Ähnlich der Geometriedaten ergibt sich eine Vielzahl von möglichen Dateiformate abhängig von der Disziplin und der verwendeten Simulationssoftware für die Beschreibung des Verhaltensmodells. Besonders im Bereich der Co-Simulation wird jedoch eine universelle Schnittstelle gefordert. Aus diesem Grund wurde bereits im Jahr 2010 das *Functional Mockup Interface (FMI)* von *Modelica Association* definiert. Die Grundlage von diesem Interface beruht auf C-Code, welcher universell eingesetzt werden kann. Aktuell liegt dieses Interface in der Version 2.0.3 (November 2022) vor und wird bereits von über 100 Tools unterstützt. Der Quellcode ist dabei OpenSource und wird unter der *2-Clause BSD* Lizenz vertrieben.

2.5.3. Code Snippets and Examples

Im Bereich der Steuerungstechnik mithilfe von SPS ist vor allem die EN 61131 der Grundpfeiler für die verwendeten Programmiersprachen. Diese Norm beruht zu einem großen Teil auf der Organisation *PLCoopen*, welche sich zum Ziel gesetzt hat die Effizienz bei der Erstellung von Steuerungssoftware zu steigern und dabei plattformunabhängig zu sein. Die großen SPS-Hersteller stützen sich dabei auf diese Norm und bieten Schnittstellen für die definierten Programmiersprachen. Damit kann der Aufwand beim Austausch der Software minimiert werden.

2.5.4. Documentation

Nur mit einer guten Dokumentation stellt eine ordnungsgemäße Verwendung eines Produktes sicher. Diese sollte leicht für den Menschen lesbar und verständlich sein. Hierzu bieten sich verschiedene Dateiformate an wie: *Portable Document Format (PDF)*, *Markdown (MD)* oder *Hypertext Markup Language (HTML)*. Jedes Format bietet andere Vorteile wie etwa eine einfache Handhabung, direkte Git-Integration oder auch eine hohe Kompatibilität über verschiedene Geräte.

2.5.5. Exchange Libraries

3. Proposed Data Structure for universal Data Exchange

3.1. Structure

3.2. Advantages and Disadvantages

4. Exemplary Application using the proposed Data Structure

4.1. Introduction

4.2. Module Separation

4.3. Module Filling

4.4. Visualization

5. Results and Evaluation

6. Summary and Outlook

6.1. Summary

6.2. Outlook

Bibliography

List of Figures

List of Tables

List of Code

List of Acronyms

COLLADA	CollAaborative Design Activity
FMI	Functional Mockup Interface
HMI	Human-machine interface
HTML	Hypertext Markup Language
IGES	Initial Graphics Exchange Specification
MD	Markdown
PDF	Portable Document Format
STEP	Standard for the Exchange of Product model data

A. Example Appendix