Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 51 (2020) 932–937

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021)
15-18 June 2021, Athens, Greece.

Formalization of engineering knowledge for industrial robots using Knowledge Fusion language

Andrei Lobov*, Tuan Anh Tran, Sarah Ann Oxman Prescott

Norwegian University of Science and Technology, Richard Birkelands Vei 2B, Trondheim 7034, Norway

* Corresponding author. Tel.: +47-45-913-455. *E-mail address:* andrei.lobov@ntnu.no

Abstract

Knowledge-based engineering aims to automate product design processes. Researches in manufacturing try to find optimal way to reduce the time required to go from intangible idea to a product in hands of a customer. With the advances of Information and Communication Technologies, it becomes possible to integrate and use information generated at different phases of a product lifecycle. That includes deciding on how robots should operate, which have got a key role in automation of manufacturing processes. Use of formalized representation of engineering knowledge for industrial robots' application can allow for better decision making. This paper proposes an approach for integrating product design with robot applications using Knowledge Fusion language. The Knowledge Fusion allows to capture engineering knowledge in computer-processable format that can be later used for automating design and manufacturing processes. An approach is illustrated with Siemens NX tools framework.

Keywords: knowledge-based engineering; product design; robotics; manufacturing.

1. Introduction

Robots can support different processes at the production floor. Examples of usage are i) machine tending, e.g. loading and unloading CNC machines; ii) performing various material handling tasks, e.g. for packaging or quality inspection; iii) performing assembly operations, e.g. picking and placing components or subassemblies to the right position; and iv) supporting manufacturing processes with a specific tool, e.g. for joining operations holding a screwdriver or a welding gun.

Robot vendors develop robot programming interfaces or software suites to simplify the engineering process of a robot work cell. These suites are often built with focus on application for specific robots with little capability to develop and share outside the lessons learnt via a limited number of robotics projects.

On the product design side, a Computer-aided Design (CAD) approach can be used to go from intangible idea of a product to its model that can be used as a source of information throughout product lifecycle. A corresponding CAD software tool can be used for a sequential modification of 3D bodies with basic operations on those such as Unite, Subtract, Extrude, etc., towards a desired outcome – a 3D model of a product that besides promoting a better understanding as a visual aid, can also be checked for some desired properties, e.g. behavior with respect to acting forces, materials, fatigue or the application of different manufacturing processes modelled in a Computer-aided Manufacturing (CAM) package often accompanying CAD software.

Also, for CAD/CAM software packages, the engineering knowledge often remains with the human engineers, i.e. in non-formal, non-machine processible representation.

A knowledge-based engineering (KBE) paradigm that

10.1016/j.promfg.2020.10.131

emerged in the 1980s [20] tries to formalize the knowledge as a set of engineering rules that can be processed by a computer, for example, in parallel with a CAD system to suggest better design options. Various languages have emerged for engineering knowledge representation. Reference [1] lists the following languages used today, including GDL, KNEXT, KnowledgeWare, Autodesk Intent, Knowledge Fusion, AML, Design++.

Still, the acceptance and wide use of KBE and the corresponding languages can be seen as a challenging task. Partially, in the case of CAD, it can seem counter-intuitive and going against the original and very purpose of visual representation and solution development by a human manipulating objects in 3D space, asking often the very same type (by profession) of humans to start writing rules in text shifting from graphical objects manipulations to sometimes long lines of text. But, in order to bring in sufficient computational resources to automate product design and help humans navigate design space, explore product alternatives, some computer-processable languages are required.

An engineer and professor W. Edwards Deming has the following quote: “It is not necessary to change. Survival is not mandatory” [21]. In some cases, like for big enterprises / complex product developers there was no choice other than to be early adopters of KBE [1]. However, nowadays it is going further as small and medium enterprises start to develop and use their KBE solutions. One global trend supporting it, is an appearance of open tools and software modules that can provide sophisticated functionality and can be retrieved from an online repository within few seconds to any computer connected to the Internet with a simple command. For example, there are finite element analysis libraries retrievable for Python programming language*. On the other hand, such global initiatives as Industry 4.0 and corresponding reference architecture model (RAMI) Industrie 4.0 framework [2] promote an integrated vision of product/system lifecycle, value chain, levels of an enterprise and production system.

This paper focuses on the use of Knowledge Fusion language supported by Siemens PLM software. Knowledge Fusion is an interpreted and object-oriented language. It “allows you to add engineering knowledge to a part by creating rules which are the basic building blocks of the language” [22]. In this work, the language is used to develop classes (rules) to support industrial robot applications. The application example is given with the welding process.

The rest of the paper is organized as follows. After the second section discussing the state of the art comes the description of our proposed approach. In the fourth section a modelling example for the described framework is presented. Conclusions and future work are then outlined in the last section.

* https://pypi.org/

2. State of the art

2.1 KBE frameworks

Active attempts to improve knowledge-based engineering (KBE) practices started already in the 1980s. Reference [8] reports on database support, which is essentially an organization, storing and management of design rules, which are expressed in the Marvel Strategy Language, mentioning of which at a given moment is found 281 times by Google search. As in many problem domains, some languages become more used than others. Knowledge Fusion used with Siemens NX is found via open search about 7900 times at the moment. If to compare with general purpose languages, such as, for example, Java and C#, having billion or tens of millions of mentionings, it could reflect that the languages to support knowledge-based engineering are not used often. Maybe, it could be the reason for existing CAD/CAM tools to start providing Application Programming Interfaces (APIs) for the frequently used general purpose programming languages. For instance, with Siemens Open NX one can choose to develop with Visual Basic, C#, C++, Java or Python.

However, although the use of general-purpose programming languages gives a great freedom to express engineering rules, the very same freedom could be an obstacle when integrating various designs or design phases executed by different engineers or teams. Having a lot of freedom, one can develop a solution in selected object-oriented programming languages using an arbitrary class structure and relationships between those. Common guidelines are missing. These can be attained based on selected knowledge-representation languages. The downside for the latter is absence of well-established IDEs in comparison to the general-purpose widely-used programming languages.

Nevertheless, the quest of improving KBE tools framework continues. Reference [11] reports the view from the 1990s for developing of the Boeing 737 project, where “at regular intervals information from each member of the design team was sent electronically to Seattle and combined into a single large coordination master model in Boeing’s Catia CAD system (IBM).” Nowadays such an integration can be automated and, again, it could be important that engineers use the same knowledge-representation languages. Wallace [10] has presented a Fastener KBE Architecture tested for Jeep Cherokee. Among other things, it has allowed for model-checking to perform faster going from 60 to 1-2 minutes. Besides the architectures for KBE, it is also important to understand the roles of engineers involved in product or system development. Reference [6] highlights the integration of domain expert, knowledge engineer and application developer having them discuss together rather than in any sequential manner. In [3], a workflow is presented along the KBE platform architecture used for design of armed vehicles. Three main knowledge bases are defined focusing sequentially on innovation design, reuse design and typical problem solving. Web Ontology Language (OWL) is used to support knowledge representation.

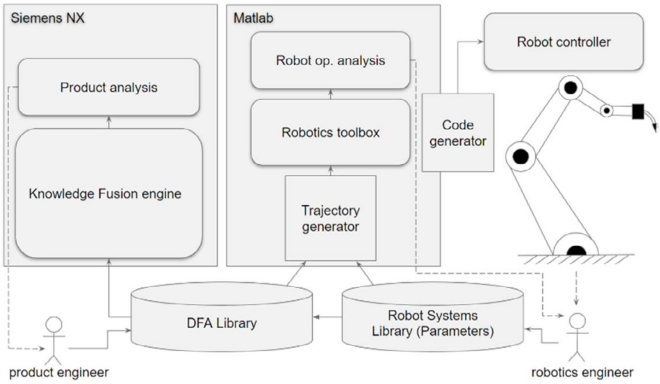

Fig. 1. Integration framework

Another example of engineering knowledge by using web standards and recommendations (the OWL is one of those) is reported in [19].

An example of a practical problem solved by employing rules expressed using Fuzzy Logic is presented in [14]. The increase of wear is predicted based on presence of water. The Fuzzy Logic is usually employed to tackle uncertainties, which often appear in engineering processes of complex products requiring consideration of many factors.

Different types of logics (e.g. Fuzzy) or methods are not always directly supported by knowledge-representation languages. However, these can be added via implementations using general-purpose programming languages or interfaces to integrate inputs from external modules.

Ligang and Sgenbo [9] report on using Knowledge Fusion language to permit designers "to take advantage of engineering knowledge to realize intelligent design on KBE." That work outlines a development processes containing editing of DFA files containing engineering rules representation in the Knowledge Fusion language.

It may be important to, not only have a way to represent the design rules, but also a way to integrate and guide the whole design process. Reference [13] a mathematical representation of the process design. A graph is created to represent and plan design activities. A process-oriented design promoted by the authors is a center element integrating design environment (system information), product data model environment (geometry data) and knowledge-based system environment that can be seen as an AI assistant to add new engineering knowledge.

As discussed in [5], even that medium-sized and small-scale companies may have powerful CAD tools, the potential of the tools is used just rudimentarily, so an approach is proposed to increase the efficiency via a proper introduction and use of a KBE system. Experimental results have shown a clear win in performance by a group, which was properly introduced to and used the KBE system.

2.2 Robotics frameworks

Robot Operating System (ROS) is a collection of software libraries and tools to support robot software development [15]. Matlab has a dedicated ROS Toolbox [16] to interface with the applications supporting ROS.

On the commercial side, many robot vendors have their own frameworks and tools supporting robot code development, simulation, offline programing, etc. These include Robotstudio by ABB[†], Stäubli Robotics Suite by Stäubli[‡] or Smart Series suite by Yaskawa[§] to name a few. If one were to check [17], s/he would find ROS developments already made 2 out of 3 mentioned industrial robots' manufacturers.

3. Approach

Figure 1 shows the suggested approach for automated design of product and development of corresponding robot code. There are two major roles, namely product engineer and robotics engineer, both are responsible for the corresponding domains.

A product engineer works with Knowledge Fusion language (DFA files) to formalize available knowledge on product design. That knowledge can be extended by parameters of the envisioned robot system to support manufacturing of the product. Kinematics parameters of the manipulator and the tool (e.g. a welding gun) can be used to infer the working envelope for the robot leading to decisions of whether a given product configuration is possible to make (e.g. the welding gun can reach and work without obstacles in all required positions for application of a welding process).

A robotics engineer can provide input by discussing and specifying robots to be used. Ideally, an input about robot parameters could also come automatically from an online repository.

[†] https://www.directindustry.com/prod/abb-robotics/product-30265-1882653.html

[‡] https://www.staubli.com/en/robotics/product-range/robot-software/

[§] https://www.motoman.com/en-us/about/media-center/news/april-2019/yaskawa-smart-series-product-line-is-ideal-for-new

```
(Child) extrusion_weld: {
 Class, ug_extruded;
 Profile, {line_1:, line_2:, line_3:};
 Direction, Vector(1,0,0);
 Start_Limit, 0;
 End_Limit, l2:;
};
```

```
(Child) swept_weld: {
  Class, ug_swept;
  Guide, {{forward, weld_path:}};
  Section, {{forward, weld_profile:}};
  Scaling, {scale_constant, 1};
  Orientation, {orientation_fixed};
};
```

Fig. 2. *extrusion_weld* subclass of *ug_extruded* for linear (on the left) profile and curved path shown by *swept_weld* subclass of *ug_swept* (on the right).

An example of such human-readable repository can be found here [17]. DFA files should be extended to include parameters of a robot system, then using Knowledge Fusion engine in Siemens NX those can be analyzed to check if a product can be made given the selected manufacturing system configuration and product requirements. The very same DFA files can be processed for input to the Trajectory Generator, which then can be analyzed with the robotic toolbox to finetune for the best performance of a robot manipulator. Finally, a code can be generated (e.g. via ROS interface) to control selected manipulator. Still, the robotics engineer may need to finetune the final code. S/he is still a key person also for updating Robot Systems Library with practical knowledge on selected industrial manipulator that can again be used to improve DFA Library for future products.

4. Example

A walkthrough of the framework presented in Figure 1 is first illustrated with a simple welding operation for the two plates.

Figure 2 shows representation of two welded pates made with Knowledge Fusion (KF) language. The ug_extruded class of the KF was used to form a graphical representation of the weld between the plates. The ug_body class was used to highlight the weld with a selected color (dark green). However, the parameters provided for instantiation of ug_extruded are of importance for communicating the weld path. These parameters are Profile (represents with what shape the weld is visualized, e.g. a triangle, but also contains the reference and coordinates of corresponding lines forming the triangle), Direction (a vector showing the direction of extrusion), Start_Limit (from which position w.r.t. profile elements to start the extrusion; usually is 0), End_Limit (how far to extrude). The units can be fractions of inches or that of millimeters, giving on models' side enough precision as required of a welding process.

For the straight-line welding operations, here the Start_Limit and End_Limit parameters can be inputs for defining the cartesian trajectory of a welding gun. These inputs are however not viable for curved weld extrusions, which instead requires a predefined curve trajectory path to follow. The start and end points are then determined by the characteristics of the trajectory path. This path parameter can become primary data for robot trajectory generation.

The curved profile is illustrated in Figure 4. In this case, the weld extrusion is defined using the Guide parameter from the ug_swept class.

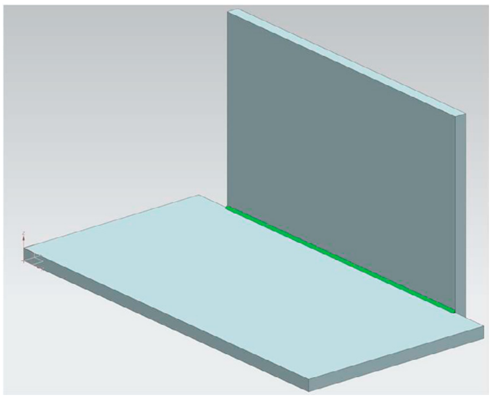

Fig. 3. Illustrating 2 welded plates.

The line_1: or the weld_path: in Figure 3 are other nested classes for the design problem containing specific parameters such as start coordinate and end coordinate for the line (ug_line) or the start and end angles, curve radius and the center point for the curves (represented with the ug_arc class in KF).

The KF supports different output formats including eXtendable Markup Language (XML). The XML files is one of the options for input to MATLAB and the Robotics Toolbox [18] for generating a trajectory for selected robot and analyzing it by robotics engineer. However, direct parsing of DFA files in MATLAB is feasible and will be illustrated below. The ROS toolbox can be used to provide interface for selected robot.

Figure 5 shows a representation of the Yaskawa GP25-12 robot with the Robotics Toolbox. The toolbox supports two basic commands jtraj and ctraj for generating joint space and cartesian space trajectories for a robot manipulator. Information on welding paths is extracted from KF classes to generate robot move trajectories to perform the welding operation. The robotics engineer can observe and analyze the trajectory within the MATLAB.

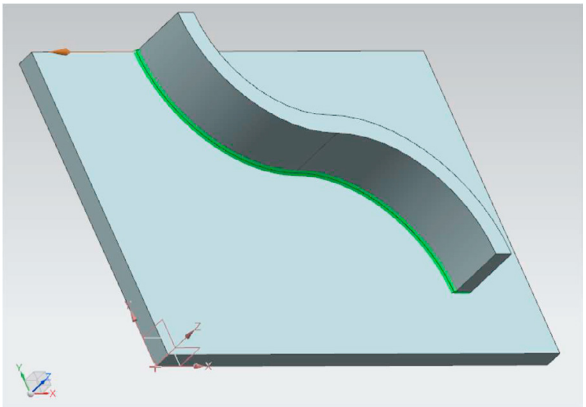

Fig. 4. Illustrating curved welding path.

In order to illustrate the parsing of DFA files in MATLAB, let us consider parsing of *ug_extruded* class instance (Fig. 2 on the left). Start_Limit and End_Limit are unique names of attributes that can be used to look up for the corresponding strings in the file and then to extract the value. There are two possible outcomes of value extraction. The value can either come as a constant, e.g 0 for Start_Limit or as a reference – l2: for given example in End_Limit. Fig. 6 shows a piece of code that first reads from the Knowledge Fusion DFA file; then extracts corresponding strings for the limit values; tries to convert those to the numbers simultaneously checking if not a number was read, i.e. reference to a variable; for the variables, it extracts the values of those. These values are used later to generate inputs for robot trajectory commands.

The object-oriented features of the Knowledge Fusion are used to find and extract required information about weld path. DFA files give a possibility to write a parser to extract relevant features to generate robot trajectory as shown in Fig. 1. The trajectory generator in the figure can be implemented using Matlab that reads DFA files (Fig. 6) supplies extracted information to the Robotics toolbox and ROS for the specific robot code generation.

5. Conclusions

The paper has presented a use of Knowledge Fusion language for KBE applications to consider and define robot operations at the product development phase for product manufacturing. Robot paths can be represented with the classes in Knowledge Fusion. The approach provides support to robotics engineers and allows integration of their knowledge at the product design phase. The MATLAB was selected for the robotics side as it is a generic, widely used tool that is also aiming at providing interfaces to different robots via the ROS.

As a future development, the framework can be extended by a tool to automatically translate and visualize the work envelope of an industrial robot using KF language. Also, other engineering knowledge representation languages should be considered to potentially be able to integrate between different KBE frameworks.

Acknowledgements

The work is supported by the Knowledge-building Project for Industry – MAROFF; Project Number: 90444802.

References

[1] Sobieszczanski-Sobieski, Morris and van Tooren, "Multidisciplinary Design Optimization Supported by Knowledge Based Engineering", Wiley, ISBN 9781118492123, 2015

[2] C. Mosch, "RAMI 4.0 and Industrie 4.0 components", VDMA, Available online: https://industrie40.vdma.org/en/viewer/-/v2article/render/15557415, [Accessed: May 2020]

[3] G. Wang, Y. Yan, L. Hu, X. Zhang and L. Wang, "Construction of knowledge based engineering platform for armored vehicle," 2009 IEEE International Conference on Industrial Engineering and Engineering Management, Hong Kong, 2009, pp. 1052-1056. doi: 10.1109/IEEM.2009.5372938

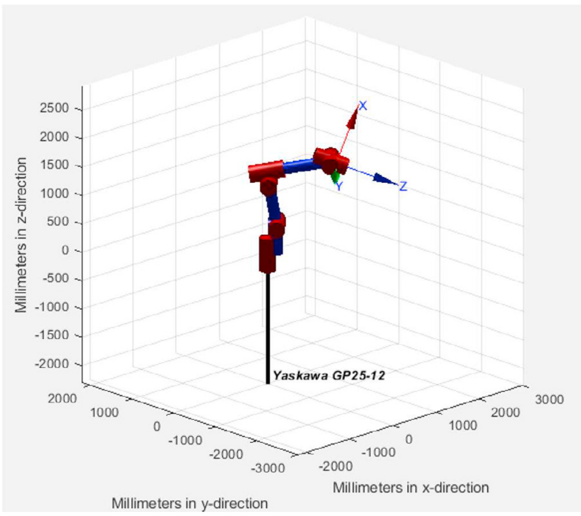

Fig. 5. Yaskawa GP25-12 robot (on the left) representation in MATLAB Robotics Toolbox (on the right).

```
% Open DFA file
fileID = fopen('kbe_weld_191010.dfa','r');
% Define format
formatSpec = '%s';
% Get file data
fileData = fscanf(fileID,formatSpec);
fclose(fileID);

% Parse string in Matlab to get values for the weld limits
Start_Limit = extractBetween(fileData,"Start_Limit,",";");
End_Limit = extractBetween(fileData,"End_Limit,",";");


% Read the values for weld start and end as numbers. Check if variable (not a number)
% was read (tfn).
[weld_start, tf1] = str2num(Start_Limit{1})
[weld_end, tf2] = str2num(End_Limit{1})

% Read the value of the variable, if conversion to number has failed.
if not(tf1)
    Start_Limit = extractBetween(fileData,Start_Limit{1},";");
    weld_start = str2num(Start_Limit{1})
end

if not(tf2)
    End_Limit = extractBetween(fileData,End_Limit{1},";");
    weld_end = str2num(End_Limit{1});
end
```

Fig. 6. Processing DFA file to read start and end of the weld for the *ug_extruded* class

[4] C. Kalavrytinos and O. I. Sievertsen, "A knowledge-based engineering approach for offshore process plant design," 2014 International Conference on Engineering, Technology and Innovation (ICE), Bergamo, 2014, pp. 1-6. doi: 10.1109/ICE.2014.6871540

[5] T. Hagenreiner, G. Engelmann and P. Köhler, "Implicit learning by means of a Knowledge-Based Engineering System," 2015 IEEE Global Engineering Education Conference (EDUCON), Tallinn, 2015, pp. 221-226. doi: 10.1109/EDUCON.2015.7095974

[6] C. Kalavrytinos, I. Marthinusen, T. Welo and O. I. Sivertsen, "Using factors associated with the lean concept in PD for identifying potential for design automation (Knowledge Based Engineering)," 2015 IEEE International Conference on Engineering, Technology and Innovation/ International Technology Management Conference (ICE/ITMC), Belfast, 2015, pp. 1-7. doi: 10.1109/ICE.2015.7438680

[7] G. Guan and Q. Yang, "Ship Structural Members Design and Optimization Based on Knowledge-Based Engineering," 2016 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), Wuhan, 2016, pp. 268-273. doi: 10.1109/ICIICII.2016.0071

[8] G. E. Kaiser, N. S. Barghouti, P. H. Feiler and R. W. Schwanke, "Database support for knowledge-based engineering environments," in IEEE Expert, vol. 3, no. 2, pp. 18-23, Summer 1988. doi: 10.1109/64.2102

[9] T. Ligang and Y. Shenbo, "The Development and Application of User Domain Parts Library on Knowledge Based Engineering," 2011 Third International Conference on Measuring Technology and Mechatronics Automation, Shangshai, 2011, pp. 765-768. doi: 10.1109/ICMTMA.2011.764

[10] G. S. Wallace, "The introduction of knowledge based engineering for design for manufacture in the automotive industry," IEE Colloquium on Successful Cases of Integrated Product Design with Manufacturing Technology (DIgest No: 1997/168), Birmingham, UK, 1997, pp. 7/1-7/5. doi: 10.1049/ic:19970907

[11] R. P. G. Barlow, G. N. Fisher and S. N. Papantonopoulos, "The application of a knowledge-based engineering system to the planning of airport facilities-a case study," IEE Colloquium on Knowledge-Based Approaches to Automation in Construction, London, UK, 1995, pp. 1/1-1/4. doi: 10.1049/ic:19950831

[12] F. Tian and M. Voskuijl, "Mechatronic Design and Optimization Using Knowledge-Based Engineering Applied to an Inherently Unstable and Unmanned Aerial Vehicle," in IEEE/ASME Transactions on Mechatronics, vol. 21, no. 1, pp. 542-554, Feb. 2016. doi: 10.1109/TMECH.2015.2441832

[13] P. W. Lin, P. Xi, R. Chen and B. Zhang, "The research of process-oriented knowledge-based CAD system," 2011 International Conference on Consumer Electronics, Communications and Networks (CECNet), XianNing, 2011, pp. 4548-4551. doi: 10.1109/CECNET.2011.5769065

[14] R. M. Parkin, "The use of knowledge-based engineering in the mechatronic embodiment of intelligent machines and systems," 1998 Second International Conference. Knowledge-Based Intelligent Electronic Systems. Proceedings KES'98 (Cat. No.98EX111), Adelaide, SA, Australia, 1998, pp. 41-50 vol.1. doi: 10.1109/KES.1998.725826

[15] The Robot Operating System (ROS), Available online: https://www.ros.org/, [Accessed: October 2019]

[16] MathWorks, "ROS Toolbox", Available online: https://www.mathworks.com/help/ros/index.html, [Accessed: October 2019]

[17] "ROS robots", Available online : https://robots.ros.org/, [Accessed: October 2019]

[18] P.I. Corke, "Robotics, Vision & Control", Springer 2017, ISBN 978-3-319-54413-7.

[19] Lobov, A.., Martikkala, A., Biswas, P. "Towards a Web-based Framework for Computer-aided Manufacturing", The International Forum on Instrumentation Engineering, Electronics and Telecommunications, Izhevsk, Russia, 2019, pp. 59-65.

[20] Chang, T.-C., Wysk, R. A., Wang H.-P., "Computer-Aided Manufacturing", Pearson, 3rd edition, 2006, ISBN: 978-0131429192

[21] Deming, E., "Out of the Crisis", MIT Press, Cambridge, MA, 1982, p. 227.

[22] Siemens, "Siemens Documentation: Programming tools", Available online: https://docs.plm.automation.siemens.com/tdoc/nx/11/nx_api/ [Accessed: June 2020]