# IMMM 2022

The Twelfth International Conference on Advances in Information Mining and Management

ISBN: 978-1-61208-960-7

June 26th –30th, 2022

Porto, Portugal

**IMMM 2022 Editors**

Petre Dini, IARIA, USA/EU

# IMMM 2022

# Forward

The Twelfth International Conference on Advances in Information Mining and Management (IMMM 2022), held between June 26th and June 30th, 2022, continued a series of events focusing on advances in all aspects related to information mining, management, and use.

The amount of information and its complexity makes it difficult for our society to take advantage of the distributed knowledge value. Knowledge, text, speech, picture, data, opinion, and other forms of information representation, as well as the large spectrum of different potential sources (sensors, bio, geographic, health, etc.) led to the development of special mining techniques, mechanisms support, applications and enabling tools. However, the variety of information semantics, the dynamic of information update and the rapid change in user needs are challenging aspects when gathering and analyzing information.

We take here the opportunity to warmly thank all the members of the IMMM 2022 technical program committee, as well as all the reviewers. The creation of such a high-quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to IMMM 2022. We truly believe that, thanks to all these efforts, the final conference program consisted of top-quality contributions. We also thank the members of the IMMM 2022 organizing committee for their help in handling the logistics of this event.

We hope that IMMM 2022 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of Information Mining and Management.

**IMMM 2022 Chairs**

**IMMM 2022 Steering Committee**

An-An Liu, Tianjin University, China

**IMMM 2022 Publicity Chairs**

José Miguel Jiménez, Universitat Politècnica de València (UPV), Spain
Laura Garcia, Universitat Politècnica de València (UPV), Spain

# IMMM 2022
# Committee

## IMMM 2022 Steering Committee

An-An Liu, Tianjin University, China

## IMMM 2022 Publicity Chairs

José Miguel Jiménez, Universitat Politècnica de València (UPV), Spain
Laura Garcia, Universitat Politècnica de València (UPV), Spain

## IMMM 2022 Technical Program Committee

Akhlaq Ahmad, Umm Al Qura University, Saudi Arabia
Zaher Al Aghbari, University of Sharjah, UAE
Stelios Andreadis, Information Technologies Institute (ITI) | Centre of Research and Technology Hellas (CERTH), Greece
Kiran Kumar Bandeli, Walmart Inc., USA
Nadezda Chalupova, Mendel University in Brno, Czech Republic
Despoina Chatzakou, Centre for Research and Technology Hellas, Greece
Seongah Chin, Sungkyul University, South Korea
Tommy Dang, Texas Tech University, USA
Qin Ding, East Carolina University, USA
Ahlem Drif, Farhat Abass University, Sétif 1, Algeria
Hannes Fassold, JOANNEUM RESEARCH - DIGITAL, Graz, Austria
(David) Dagan Feng, The University of Sydney, Australia
Paolo Garza, Politecnico di Torino, Italy
Alessandro Giuliani, University of Cagliari, Italy
David Griol, University of Granada, Spain
Gabriel Henrique de Souza, Federal University of Juiz de Fora, Brazil
Tzung-Pei Hong, National University of Kaohsiung, Taiwan
Yin-Fu Huang, National Yunlin University of Science and Technology, Taiwan
Muhammad Nihal Hussain, University of Arkansas at Little Rock, USA
Liliana Ibeth Barbosa-Santillan, University of Guadalajara, Mexico
Young-Gab Kim, Sejong University, South Korea
Cristian Lai, ISOC - Information SOCiety | CRS4 - Center for Advanced Studies, Research and Development in Sardinia, Italy
Jean-Charles Lamirel, Université de Strasbourg Equipe SYNALP (ex. INRIA TALARIS) - LORIA - Nancy, France
Mariusz Łapczyński, Cracow University of Economics, Poland
Yuening Li, Texas A&M University, USA
Chih-Wei Lin, Fujian Agriculture and Forestry University, China
An-An Liu, Tianjin University, China
Flaminia Luccio, Università Ca' Foscari of Venice, Italy
Francesco Marcelloni, University of Pisa, Italy
Michele Melchiori, Università degli Studi di Brescia, Italy

Vasileios Mezaris, Information Technologies Institute (ITI) | Centre for Research and Technology Hellas (CERTH), Thessaloniki, Greece
Mohit Mittal, INRIA, France
Jose Manuel Molina Lopez, University Carlos III de Madrid, Spain
Muhammad Ali Nayeem, Bangladesh University of Engineering and Technology, Bangladesh
Naoko Nitta, Osaka University, Japan
M Solel Rahman, Bangladesh University of Engineering & Technology (BUET), Bangladesh
Takfarinas Saber, University College Dublin, Ireland
Andreas Schmidt, Karlsruhe University of Applied Sciences & Karlsruhe Institute of Technology (KIT), Germany
Harsh Shrivastava, Singapore University of Technology and Design, Singapore
Josep Silva, Universitat Politècnica de València, Spain
Alvaro Suarez Sarmiento, Universidad de Las Palmas de G.C., Spain
Tatiana Tambouratzis, University of Piraeus, Greece
Abdullah Uz Tansel, Baruch College CUNY, USA
Edgar Tello-Leal, Universidad Autónoma de Tamaulipas, Mexico
Lorna Uden, Staffordshire University, UK
Paula Viana, Polytecnic of Porto - School of Engineering & INESC TEC, Portugal
Maria Luisa Villani, Italian National Agency for New Technologies, Energy and Sustainable Economic Development (ENEA), Italy
Ning Wu, School of Computer Science and Engineering - Beihang University, China
Guang Yang, Walmart, USA
Shibo Yao, New Jersey Institute of Technology, USA
Tong Zhao, Amazon, USA
Dongqing Zou, Sensetime, Hong Kong

**Copyright Information**

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

# Table of Contents

# Semantic Support for DSL Demonstrator in an Additive Manufacturing Environment

Fathia Bettahar

Capgemini Engineering
Toulouse, France
e-mail : fathia.bettahar@capgemini.com

*Abstract*— **In this work, we describe a Domain-Specific Language (DSL) demonstrator that controls the interaction with the user in an additive manufacturing environment. Its objective is to give appropriate answers to queries asked by a user. Several modules of the environment are used to bring theses answers: the DSL grammar, which lets users query the components of a complicated system without having to learn an unfamiliar query language; the user interface, which can be adaptable according to user needs; and the graph database management system developed to parse the knowledge base. All the modules share a central ontology describing the additive manufacturing domain. Our ontology is used both for building a formal grammar and as knowledge base, providing concept definitions.**

*Keywords-Ontology; Domain-Specific Language; Additive Manufacturing.*

## I. INTRODUCTION

The development of complex software systems within multidisciplinary sub-teams raises new software engineering challenges. It is important to understand complex problems and to give appropriate solutions. Model-Driven Engineering (MDE) [1] is the approach proposed to solve these challenges. It raises the level of abstraction in traditional programming languages by using models. MDE focuses on the use of models to understand, analyze, and develop complex software system. MDE advocates the description of a system by a set of multiple Domain-Specific Languages (DSLs) [2].

DSLs, as opposed to General Programing Languages (GPLs), are designed to describe a program at the adequate level of abstraction. DSLs narrow the gap between a problem domain and its implementation. Tools and collaboration of DSLs are among important research challenges [3] in model-driving engineering. Particular challenges include tools for defining and composing domain-specific languages. Usually, different domain specific languages are used simultaneously to define the system from several viewpoints, hence the necessity of semantic interoperability and collaboration between them.

In this work, we propose an approach based on ontology able to define DSL and which aims at a semantic interoperability between DSLs in a complex software system. The objective of this approach is to foster communication and collaboration among team members (developers, domain experts and end-users). This objective can be achieved since ontologies give a common semantic representation of the domain modeled, which can be shared by DSLs in Model-Driven Software Engineering (MDSE).

In this paper, we propose a method to build a DSL in the context of the additive manufacturing domain. We define a Domain Specific Language describing the user Request (DSLReq) from Additive Manufacturing Ontology (AMO). *The focus of this paper is to present a semantic support DSL demonstrator in Additive Manufacturing (AM) domain.* Thus, our approach produces a grammar useful to parse an additive manufacturing knowledge base represented in a semantic graph.

The rest of the paper is organized as follows: In Section II, we present the research work related to our approach. In Section III, we present our grammar, its functionality and introduce its rules in part B. We also discuss the role of ontology in our approach in part A. In part C, we give a scenario example to explain our developed system. The paper ends with some concluding remarks and some guidelines for future work.

## II. RELATED WORK

To understand the ontology role, we can refer to the ontology definition of Gruber [4] "explicit specification of conceptualization", where conceptualization is "a set of objects, which an observer thinks exist in the world of interest and relations between them". So, ontology O related to a domain D is a set of the concepts ® and relations ® among them as in (1).

$$O_D = \{ C, R \}_{C \in D}$$

(1)

Each ontology concept is defined by some properties and axioms. This definition makes ontologies very expressive and powerful means for domain modeling, and they are proposed as one of the approaches to create Domain Specific Languages (DSLs).

Many research works use ontologies to support the development of DSLs. Ontologies help in the initial phase of DSL development called Domain Analysis Phase [5]. In this phase, the ontology defines the domain, its terminology, the concepts, and their dependencies.

In [6], the authors proposed the OntoDSL tool that allows to use ontologies during the design phase and to guide the DSL programmer to develop an expressive language. OntoDSL provides automated reasoning services that can be

practically used by DSL designers and DSL users. In other works, researchers emphasize the advantages of using ontologies to support DSLs development. For that reason, [7] proposes an Onto2Gra framework to automatically generate the DSL grammar from an ontology. This framework provides the mapping of concepts into grammar symbols based on Context Free Grammars (CFG), and the mapping of relations into grammar productions. These approaches used ontologies to solve DSL challenges, in particular interoperability with other languages, because more than one language must be combined in the modeling of systems.

Ontologies have proved themselves in various domains as a tool for adding semantic to data and for achieving a desirable level of interoperability [6]. For that reason, we believe that ontology is the solution of both the interaction problem between heterogenous components within a complex computer system and the interaction problem between DSLs.

Our approach has the same goals as OntoDSL because it is a framework in a model-driven approach. Differently from OntoDSL our approach aims to build a DSL from an ontology able to parse the knowledge base. The result of our approach is a grammar describing a user request to get a response from the ontology and the AM domain without having to learn an unfamiliar query language.

## III. PRINCIPLE

In this work, we present our approach in additive manufacturing context. The development and industrial adoption of Additive Manufacturing (AM) technologies required the development of computer systems able to aid decision making and the data management in a homogenous way. Ontologies can answer these requirements because they have competencies [7] to both exchange data across applications and automatically reason over knowledge for decision making.

Besides, additive manufacturing systems manifest a low degree of interoperability, and this creates an interaction problem between enterprises or different branches of an enterprise. Ontologies are valuable tools for solving such problems [8]. Given that, an ontology builds on classes to represent materials, products components, process's parameters, and process's parts [9].

Our approach is based on two pillars. The first pillar is a domain specific language that allows understanding, validation, and modification of the additive manufacturing domain. The second pillar is an ontology used both for the construction of a formal grammar and knowledge base, providing concept definitions.

For that purpose, we define a framework based on ontology. This framework is structured around three interconnected models, as illustrated in Figure 1. AM ontology defines an additive manufacturing domain, a user interface, which generates a Cypher query [10] from our DSL grammar and a semantic graph database based on AM ontology. We have generated the DSLReq grammar from our ontology.
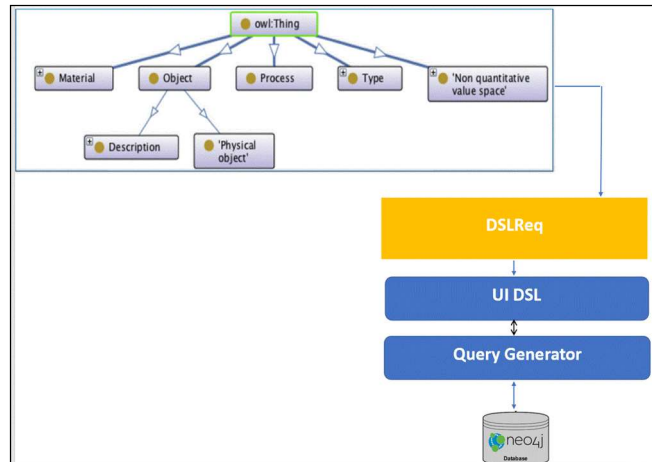


Figure 1. Approach Architecture

In this section, we discuss firstly the ontology building method. Secondly, we explain the process of DSLReq generation and the role of ontology in this process.

### A. Additive Manufacturing Ontology (AMO)

The methodology of developing an ontology is chosen according to the complexity of the domain and its requirements. For the manufacturing domain, the method MOP (Machine of a Process) [11] is used. MOP aims to reuse existent ontology to limit the development from scratch. Upper-level ontology for manufacturing was used as a reference to evaluate and to improve specific domain ontology. Thus, this method aims to hold interoperability among domain users. In order to achieve a high-level of interoperability, we developed our additive manufacturing ontology according to the research work in [9].

[9] proposes first the reuse of upper-level ontology DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) [12] to facilitate the grouping of classes sharing common high-level characteristics. Second, it proposes the extension of the upper-level branch to additive manufacturing.

The AM ontology includes broad classes and relations, which can be easily specialized to meet specific modeling scenarios and requirements: MATERIAL, NON-QUANTITATIVE VALUE SPACE, OBJECT, PROCESS and TYPE. OBJECT covers three classes: PYSICAL OBJECT, ORGANIZATION, and DESCRIPTION. PYSICAL OBJECT covers various classes, among which MFGDEVICE, PRODUCT, and FEATURE. MFGDEVICE is composed of various classes, the most important one being MFGMACHINE for the conceptualization of manufacturing machines. AMMACHINE defines an explicit model for additive manufacturing machines. AMMACHINE has a relationship with the AMPROCESS class called MecanismOf. This relationship gives all processes in which a machine can participate.

Axioms are added to explain that a machine may never be employed in any manufacturing process, for example, as defined in (2):

$$\text{MFGMACHINE} \sqsubseteq \text{MFGDEVICE} \sqcap \forall\text{mecanismOf.MFGPROCESS}$$
$$\text{AMMACHINE} \sqsubseteq \text{MFGMACHINE} \sqcap \forall\text{mecanismOf.AMPROCESS}$$

(2)

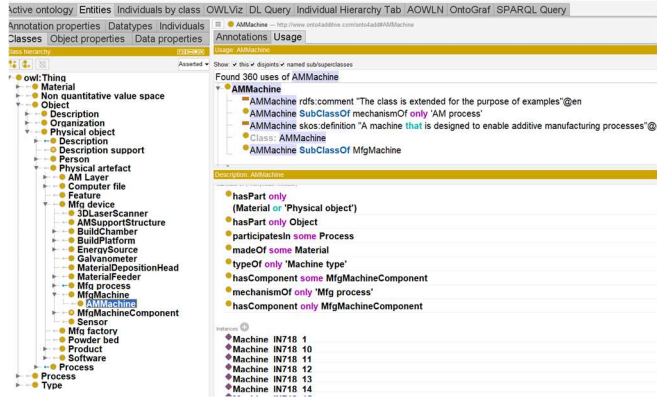Figure 2 illustrates the AMMACHINE concept and its properties in our AM ontology created with Protégé [13].



Figure 2. AMMachine Class in AMO ontology

The ontology supports the instances of concepts and relations between them. We use the Protégé plug-in to populate our ontology.

### B. From ontology to DSL grammar

Within our system, grammar advocates the description of user queries by a set of domain-specific language rules. Each rule consists of an ontology element: concept, relation, or attribute. Our approach is to build a DSL grammar able to generate a Cypher request to parse AMO. For that purpose, we define the request structure according to the ontology structure.

To convert the ontology into a grammar, we define a set of translation rules. The rules are defined in a hierarchical structure. *Get* is the super rule that connects all rules. It is composed of three sub-rules: *Type, Where* and *Return*. Firstly, the *Type* rule is the result of rule translation to map concepts and relations into the grammar. Secondly, the *Where* rule defines constraints of cardinality like RDFS range and RDFS domain within the grammar. Finally, the *Return* rule translates ontology instances corresponding to the request result (Figure 3).
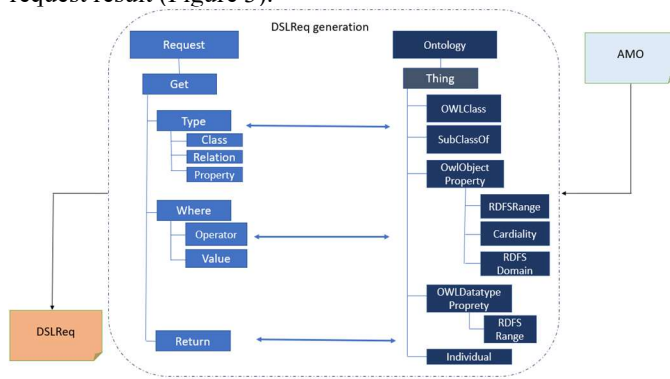


Figure 3. From ontology to DSL

Each rule is related to an element in the ontology. So, the grammar is composed of three elements related to the ontology: **Node**, **Relation** and **Property**. The grammar always starts by *Type* rule. The request takes as its input the elements in the *Type* rule, operates on them as specified in the *Where* and *Return* rules, and then produces the RDF triple researched by the user. The grammar structure is presented in rule (3):

(3)

$$Get:$$
$$\{Get\}'Get'(type = (Type))('Where'(clauses +=$$
$$Clause) *)?\, (retours = (Return))$$

The *Type* rule defines the input type of the request. An input may be a concept, a relation, or a property of the AM ontology. For that reason, the *Type* rule contains a *Node* element corresponding to an ontology concept. It also contains a *Relation* rule. The *Relation* rule is composed of the ontology relation and a range of relation called *destNode*. The *Type* rule is presented in (4), where *C, C'* are ontology concepts and *Rel* is the relation between them:

(4)

$$C\ rel\ C' \implies Type:$$
$$\{type\}(firstNode = Node)(relations +\ = RelationTo)) *$$
$$RelationTo:$$
$$(relation = Rel)(destNode = Node)$$
$$Rel:$$
$$\{Rel\}(name = ID)?':' (relationName = ID|'All')$$
$$Node:$$
$$\{Node\}(name = ID)?':'(className = ID|'All')$$

The *Where* rule defines constraints about an ontology element. The constraint is defined by an operator and its value. A constraint about an attribute is called data property in OWL [14]. We use a second rule in our grammar to define this constraint. Considering $C$, a concept of AM ontology and $p$ its data type property. We define a restriction on $p$ as follows (5):

(5)

$$p\ Domain\ C\ and\ p\ has\ value \implies Clause:$$
$$(property = ID)'of'(node = [Node])$$
$$(operator = Operator)(value = Value)$$

The *Return* rule lists the result of the query. The query result is defined in the *Return* rule, as seen in (6). A function is a facultative element, which presents any logic and mathematic functions that we can apply to a property. Our rule returns instances from the ontology according to type and constraints defined in the mentioned grammar rules in (4) and (5).

(6)

$$Return:$$
$$\{Return\}'Return'(properties += Property) +$$
$$Property:$$
$$(function = Function)?\ propertyNam = ID'of'node$$
$$= [Node]$$

The elements defined in our DSL grammar are instances of EObject of Eclipse Modeling Framework (EMF) [15] Ecore models. To implement our DSL grammar, we use the Xtext [16] tool of the EMF ecosystem.

### C. Scenario example

In this scenario example, we explain our system by considering the following user story: we seek the process list where the hatching in micrometer equals to 35. This list aims to classify processes in categories.

To answer this user story, there are four steps:

1. The user can write a simple query without having to learn a Cypher language like in (7). In our interface, we define queries templates with our DSLReq, which can help the user to autocomplete the request.

$$\textbf{\textit{Get }} n: AMMachine \ r: mechanismOf \ m: AMProcess$$
$$\textbf{\textit{Where }} hasHatcingInMicromer \ of \ m = 35$$
$$\textbf{\textit{Return }} label \ of \ m \qquad (7)$$

2. The query must have a syntactic and semantic validation. Our DSL validator generates an error if the term given by the user is not in the AM ontology.

3. The system generates a Cypher query by using the Query generator (Figure 4).

```
                                                    Generate
 AMMachine.cypher
 1  MATCH(n:AMMachine)-[r:mechanismOf]-(m:AMProcess)WHERE m.hasHatchingInMicrometer=35
 2  RETURN {m_label:m.label }
```
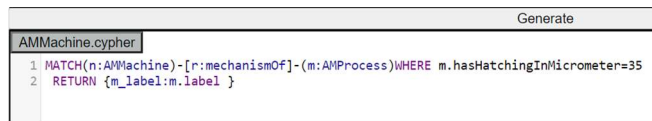
Figure 4. Cypher query

4. The system connects to the knowledge base and parses the ontology by using this request. Finally, it gives the AMProcess list appropriate to answer the user query.

### IV. CONCLUSION AND FUTURE WORK

In this paper, we proposed a query demonstrator able to give appropriate results for user requests by using a DSL grammar. All modules of our system shared a central ontology describing the additive manufacturing domain.

This ontology is used for building and validation of a DSL request. It is also used to support queries for retrieving that knowledge. However, an additive manufacturing environment requires knowledge contributions from different stakeholders, so it is necessary for software engineering to interact with other engineering disciplines. For that reason, the presented approach needs to be complemented by a set of multiple Domain-Specific Languages. Each DSL will relate to an engineering discipline and will be in interaction with the other DSLs. To achieve these results, the ontology will provide interoperability between the DSLs in our system.

### REFERENCES

[1] M. Franzago, D. D. Ruscio, I. Malavolta, and H. Muccini, "Collaborative Model-Driven Software Engineering: A Classification Framework and a Research Map," *IEEE Trans. Software Eng.*, vol. 44, no. 12, pp. 1146–1175, Dec. 2018, doi: 10.1109/TSE.2017.2755039.

[2] J. Whittle, J. Hutchinson, M. Rouncefield, H. Burden, and R. Heldal, "Industrial Adoption of Model-Driven Engineering: Are the Tools Really the Problem?" in *Model-Driven Engineering Languages and Systems*, vol. 8107, A. Moreira, B. Schätz, J. Gray, A. Vallecillo, and P. Clarke, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 1–17. doi: 10.1007/978-3-642-41533-3_1.

[3] A. Bucchiarone, J. Cabot, R. F. Paige, and A. Pierantonio, "Grand challenges in model-driven engineering: an analysis of the state of the research," *Software and Syst Model*, vol. 19, no. 1, pp. 5–13, Jan. 2020, doi: 10.1007/s10270-019-00773-6.

[4] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *International Journal of Human-Computer Studies*, vol. 43, no. 5–6, pp. 907–928, Nov. 1995, doi: 10.1006/ijhc.1995.1081.

[5] R. Tairas, M. Mernik, and J. Gray, "Using Ontologies in the Domain Analysis of Domain-Specific Languages," in *Models in Software Engineering*, vol. 5421, M. R. V. Chaudron, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 332–342. doi: 10.1007/978-3-642-01648-6_35.

[6] K. Michal, Š. Michal, and B. Zdeněk, "Interoperability through ontologies," *IFAC Proceedings Volumes*, vol. 45, no. 7, pp. 196–200, 2012, doi: 10.3182/20120523-3-CZ-3015.00039.

[7] M. von Rosing and J.A. Zachman Sr., "The Need for a Role Ontology," *IJCSSA*, vol. 5, no. 1, pp. 1–24, Jan. 2017, doi: 10.4018/IJCSSA.2017010101.

[8] M. Mohd Ali, R. Rai, N. Otte, and B. Smith, "A product life cycle ontology for additive manufacturing," *Computers in Industry*, vol. 105, pp. 191–203, Dec. 2018, doi: 10.1016/j.compind.2018.12.007.

[9] E. Sanfilippo, F. Belkadi, and A. Bernard, "Ontology-based knowledge representation for additive manufacturing," *Computers in Industry*, vol. 109, pp. 182–194, Aug. 2019, doi: 10.1016/j.compind.2019.03.006.

[10] "Cypher Query Language - Developer Guides." https://neo4j.com/developer/cypher/ (accessed Mar. 10, 2022).

[11] L. Ramos, R. Gil, D. Anastasiou, and M. J. Martin-Bautista, "Towards a Machine of a Process (MOP) ontology to facilitate e-commerce of industrial machinery," *Computers in Industry*, vol. 65, no. 1, pp. 108–115, 2014.

[12] S. Borgo *et al.*, "DOLCE: A descriptive ontology for linguistic and cognitive engineering1," *Applied Ontology*, pp. 1–25, Nov. 2021, doi: 10.3233/AO-210259.

[13] "Protégé" https://protege.stanford.edu/ (accessed Mar. 01, 2022).

[14] "OWL - Semantic Web Standards." https://www.w3.org/OWL/ (accessed Feb. 17, 2022).

[15] R. Gronback, "Eclipse Modeling Project | The Eclipse Foundation." https://www.eclipse.org/modeling/emf/ (accessed Mar. 23, 2022).

[16] "Xtext - Language Engineering Made Easy!" https://www.eclipse.org/Xtext/ (accessed Feb. 21, 2022).