Smart lifts

Slee, D.; Cain, S.; Vichare, P.; Olszewska, J.I.

Published in:
Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - KEOD

DOI:
10.5220/0010690700003064

Published: 01/01/2021

Document Version
Peer reviewed version

Smart Lifts: An Ontological Perspective

D. Slee, S. Cain, P. Vichare, and J.I. Olszewska
School of Computing and Engineering, University of the West of Scotland, UK

Keywords: Industry 4.0, Smart Lift Services, Digital Twin, Human-Machine Systems, Cyber-Physical Systems, Knowledge Engineering, Knowledge Representation, Interoperability, Ontology Engineering, Enterprise Ontology, Ontological Domain Analysis and Modeling.

Abstract: Nowadays, there is a growth of smart factories and Industry 4.0 technologies, involving Artificial Intelligence (AI) systems. These ones require interoperable solutions. In particular, ontologies have been widely used for capturing, sharing, and representing knowledge in an interoperable way, that both humans and machines can understand. Indeed, ontologies allow humans to communicate with machines in a semantic way, while machines are able to make automated reasoning about the concepts and relationships which are encoded in the ontology. For this purpose, this paper proposes the first-ever domain ontology for smart lifts. Its domain covers smart lift design, operation, and maintenance, while its scope is to aid in automating such lift services. This smart lift ontology (SLO), which contains 144 classes and 749 axioms, has been successfully developed in collaboration with the elevator industry.

1 INTRODUCTION

The use of Artificial Intelligence (AI) in our Society (Cockburn et al., 2018) is currently increasing in applications, ranging from human-centered systems (Wilding et al., 2020) to intelligent manufacturing (Lewandowski and Olszewska, 2020), expanding from Smart Cities (Costanzo et al., 2016) to Smart Factories (Xu and Hua, 2017), and contributing to the current fourth industrial revolution, or 'Industry 4.0' (I4.0) (Marr, B., 2018). This trend leads to advances in digitalization and communication as well as new manufacturing processes and innovative products (Koh et al., 2019).

AI-driven technologies used throughout I4.0 include cyber-physical systems (CPS) (Derler et al., 2011), internet of things (IoT) (Feki et al., 2013), intelligent agents (IA) (Kannengiesser and Muller, 2013), human-machine interactions (HMI) and augmented reality (AR) (Gorecky et al., 2014), autonomous robotics (Bonci et al., 2017), 3D printing, simulation and digital twin modeling (Zhong et al., 2017), cloud computing, cybersecurity, machine learning and big data (Alcacer and Cruz-Machado, 2019).

That results in *smart products*, *smart machines*, and/or *augmented operators*, but also in challenges such as interoperability, virtualization, decentralization, real-time capability, service-orientation, and modularity (Koh et al., 2019). Besides, I4.0 promotes features such as interoperability, agility, flexibility, decision-making, connectivity, quality, safety, efficiency and cost reductions (Dopico et al., 2016). In particular, interoperability refers to the ability of two systems to communicate with and understand each other (Koh et al., 2019). Moreover, there are four levels of interoperability in I4.0, namely, operational, systematic, technical and semantic interoperability (Ide and Pustejovsky, 2010). Within smart manufacturing, semantic interoperability of heterogeneous machines and/or agents, in order to be able to communicate with one another in or across smart factories, is one of the major features of I4.0 (Nilsson and Sandin, 2018). Indeed, interoperability constructs a trusted environment in a manufacturing system, in which information is accurately and swiftly shared among machines and humans, resulting in a

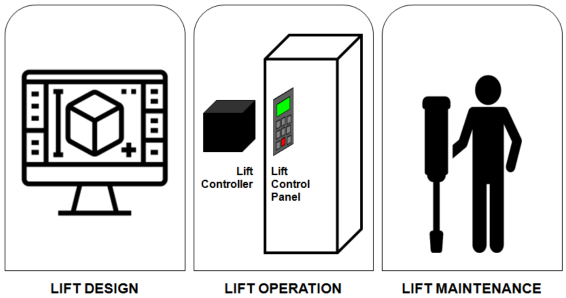


Figure 1: Overview of the lift services.

cost-saving operation with higher productivity (Koh et al., 2019).

In the case of the elevator industry, on one hand, lift manufacturing starts to embrace I4.0 technologies (Nott, 2018) and challenges (Berger, 2020) and, on the other hand, lift services enter in the age of smart buildings (Onag, G., 2019) and smart cities (Hoyes and Mair, 2020), all requiring interoperable solutions.

Hence, Knowledge Engineering (KE) techniques such as ontologies (Lee, 2019) are useful for Industry 4.0. (Sampath Kumar et al., 2019). Indeed, an ontology is a concept, which was defined by (Gruber, 1995) as an explicit specification of a conceptualization, which allows semantic interoperability (Kalyazina and Kashevnik, 2018), leading to information being easily read and interpreted by both machines and humans alike.

Therefore, ontologies have been used in Industry 4.0 for intelligent manufacturing (Xu and Hua, 2017), agent-based manufacturing (Tang et al., 2018), or cognitive manufacturing (Ferrer et al., 2019). More specific ontologies have been applied to production line (Cheng et al., 2016), micro-device assembly (Cecil et al., 2018), sensor data analysis (Gyrard et al., 2016), radio-frequency identification (RFID) system configuration (Tsalapati et al., 2021), or AI-system testing (Olszewska, 2020).

Some ontologies have been designed for cyber-physical systems (CPS) (Engel et al., 2018; Al Sunny et al., 2017; Wan et al., 2018; Brings et al., 2018; Torsleff et al., 2018; Hildebrandt et al., 2020; Voinov and Senokosov, 2021), Internet of Things (IoT) (Ma et al., 2014), Digital Twin modeling (Steinmetz et al., 2018), system of systems (SoS) modeling (Zhu et al., 2017), Web of Things (WoT) (Sujith et al., 2011), robotics (Fiorini et al., 2017), cloud robotic systems (CRS) (Pignaton de Freitas et al., 2020), human-machine interactions (HMI) (Jost et al., 2017), and Human-Robot Interactions (HRI) (Smirnov et al., 2016).

On the other hand, ontologies have been developed for critical infrastructures (Canito et al., 2020), Smart Buildings (Kunold et al., 2019), and Smart Cities (Burns et al., 2018). However, there is no existing ontology for the smart lift domain at the moment.

Thus, in this work, we endeavoured to develop such smart lift domain ontology (SLO).

The core knowledge of our smart lift ontology includes elevator manufacturing as well as lift services, as depicted in Fig. 1.

This domain ontology for smart lifts has been developed using Enterprise Ontology (EO) methodology (Dietz and Mulder, 2020), since EO is a mature ontology development methodology for industry-based domain ontologies (Fox and Gruninger, 1998; Albani and Dietz, 2007; Syamili and Rekha, 2017).

SLO ontology has been coded in Web Ontology Language Descriptive Logic (OWL DL) (Olszewska, 2021), which is considered as the international standard for expressing ontologies and data on the Semantic Web (Guo et al., 2007), and using Protege tool (Rubin et al., 2007) in conjunction with the HermiT reasoner (Glimm et al., 2014).

Thence, the resulting SLO-based intelligent system provides an interoperable solution for lift design, operation, and maintenance.

The paper is structured as follows. Section 2 presents the purpose and the building of our ontology for smart lift services (SLO), while its evaluation and documentation are described in Section 3. Conclusions are drawn up in Section 4.

2 DEVELOPED SLO ONTOLOGY

To develop the SLO ontology, we followed an ontological development life cycle (Fernandez et al., 1997; Jones et al., 1998; Bertolazzi et al., 2001; Fernandez-Lopez and Gomez-Perez, 2002; Gomez-Perez et al., 2004) based on the Enterprise Ontology (EO) Methodology (Dietz and Mulder, 2020).

The adopted ontological development methodology consists of four main phases (Olszewska and Allison, 2018), which cover the whole development cycle, as follows:

1. identifications of the purpose of the ontology (Section 2.1);
2. ontology building which consists of three parts: the capture to identify the domain concepts and their relations; the coding to represent the ontology in a formal language; and the integration to share ontology knowledge (Section 2.2);
3. evaluation of the ontology to check that the developed ontology meets the scope of the project (Section 3.1);
4. documentation of the ontology (Section 3.2).

2.1 ONTOLOGY PURPOSE

The scope of this smart lift domain ontology is (i) to provide the elevator industry with a new technological solution that copes with smart manufacturing challenges such as interoperability and (ii) to assist the relevant stakeholders with smart lift services in context of smart cities and smart buildings.

A way to refine the scope of the ontology is to sketch a list of questions called *competency questions*, that an intelligent agent based on the proposed ontology should be able to answer (Gruninger and Fox, 1995).

In the smart lift domain, the list of competency questions includes but is not limited to:

- What are the modules of the lift controller?
- Where is the electrical compartment located?
- What does the door node handle?
- What is the lift display used for?
- How the call button is connected to the key switch?
- Is the fingerprint reader optional?
- What is the TagReader RFID's part number?
- Who is the supplier of the CiVoice part?
- How to lock the lift?
- What is the rated load of the platform lift Cibes model A4000 type A5?
- How to adjust the overload switch?
- Where is the emergency stop located?
- How to emergency lower a lift?
- Which maintenance actions need to be performed in the machine area?
- How often the brakes need to be tested?

Therefore, the SLO ontology aims to contribute to the elicitation of the elevator industry knowledge and the formalization of concepts for lift services which comprise lift design, operation, and maintenance. Furthermore, the smart lift domain encompasses different types of lifts such as platform lifts, goods lifts, and passenger lifts.

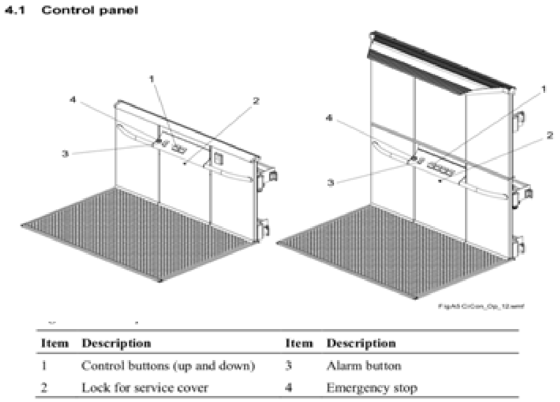
2.2 ONTOLOGY BUILDING

The ontology building consists of three parts: capture to identify the domain concepts and their relations (Section 2.2.1); coding to represent the ontology in a formal language (Section 2.2.2); and integration to share ontology knowledge (Section 2.2.3).

2.2.1 CONCEPT CAPTURE

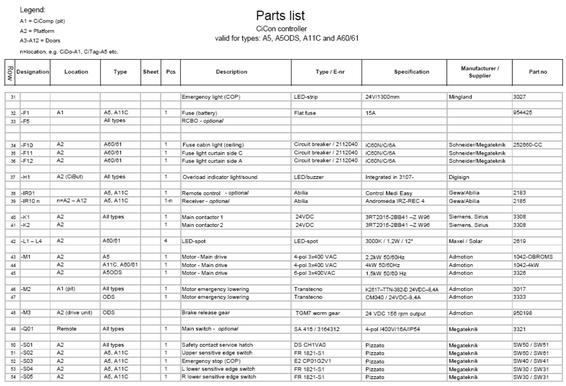
The knowledge capture consists in the identification of concepts and their relations within the elevator industry and smart lift service domains.

Hence, the SLO domain contains technical data about elevators' components and parameters that are used for manufacturing and configuration and that can be extracted, e.g., from lift documentation such as product documents, assembly instruction manuals, and installation guides (Siikonen, 1997; Hoon, 2006; Thyssen Krupp, 2014) as well as information on lift services, which consist of lift design, lift operation, and lift maintenance, from user guides, operation



Item	Description	Item	Description
1	Control buttons (up and down)	3	Alarm button
2	Lock for service cover	4	Emergency stop

Figure 2: Diagram of the control panel of the system (A5000 OM Manual).



Legend:
A1 = CiComp (pit)
A2 = Platform
A3-A12 = Doors
n=location, e.g. CiDo-A1, CiTag-A5 etc.

Parts list
CiCon controller
valid for types: A5, A5ODS, A11C and A60/61

Row	Designation	Location	Type	Sheet	Pcs	Description	Type / E-nr	Specification	Manufacturer / Supplier	Part no
31						Emergency light (COP)	LED-strip	24V/1300mm	Mingland	3027
32	-F1	A1	A5, A11C		1	Fuse (battery)	Flat fuse	15A		954425
33	-F5		All types			RCBO - *optional*				
34	-F10	A2	A60/61		1	Fuse cabin light (ceiling)	Circuit breaker / 2112040	iC60N/C/6A	Schneider/Megateknik	252660-CC
35	-F11	A2	A60/61		1	Fuse light curtain side C	Circuit breaker / 2112040	iC60N/C/6A	Schneider/Megateknik	
36	-F12	A2	A60/61		1	Fuse light curtain side A	Circuit breaker / 2112040	iC60N/C/6A	Schneider/Megateknik	
37	-H1	A2 (CiBut)	All types		1	Overload indicator light/sound	LED/buzzer	Integrated in 3107-	Digisign	
38	-IR01		A5, A11C		1	Remote control - *optional*	Abilia	Control Medi Easy	Gewa/Abilia	2183
39	-IR10 n	n=A2 – A12	A5, A11C		1-n	Receiver - *optional*	Abilia	Andromeda IRZ-REC 4	Gewa/Abilia	2185
40	-K1	A2	All types		1	Main contactor 1	24VDC	3RT2015-2BB41 –Z W96	Siemens, Sirius	3308
41	-K2	A2			1	Main contactor 2	24VDC	3RT2015-2BB41 –Z W96	Siemens, Sirius	3308
42	-L1 – L4	A2	A60/61		4	LED-spot	LED-spot	3000K / 1,2W / 12°	Maxel / Solar	2619
43	-M1	A2	A5		1	Motor - Main drive	4-pol 3x400 VAC	2,2kW 50/60Hz	Admotion	1042-OBROMS
44		A2	A11C, A60/61		1	Motor - Main drive	4-pol 3x400 VAC	4kW 50/60Hz	Admotion	1042-4kW
45		A2	A5ODS		1	Motor - Main drive	6-pol 3x400VAC	1,5kW 50/60 Hz	Admotion	3326
46	-M2	A1 (pit)	All types		1	Motor emergency lowering	Transtecno	K2617–TTN-382-D 24VDC–8,4A	Admotion	3017
47			ODS		1	Motor emergency lowering	Transtecno	CM040 / 24VDC-8,4A	Admotion	3333
48	-M3	A2 (drive unit)	ODS			Brake release gear	TGM7 worm gear	24 VDC 156 rpm output	Admotion	950198
49	-Q01	Remote	All types		1	Main switch - *optional*	SA 416 / 3164312	4-pol /400V/16A/IP54	Megateknik	3321
50	-S01	A2	All types		1	Safety contact service hatch	DS CH1VA0	Pizzato	Megateknik	SW50 / SW51
51	-S02	A2	A5, A11C		1	Upper sensitive edge switch	FR 1821-S1	Pizzato	Megateknik	SW60 / SW61
52	-S03	A2	A5, A11C		1	Emergency stop (COP)	E2 CP01G2V1	Pizzato	Megateknik	SW40 / SW41
53	-S04	A2	A5, A11C		1	L lower sensitive edge switch	FR 1821-S1	Pizzato	Megateknik	SW30 / SW31
54	-S05	A2	A5, A11C		1	R lower sensitive edge switch	FR 1821-S1	Pizzato	Megateknik	SW30 / SW31

Figure 3: Parts list for the CiCon controller system (A5-1X230 Part List).

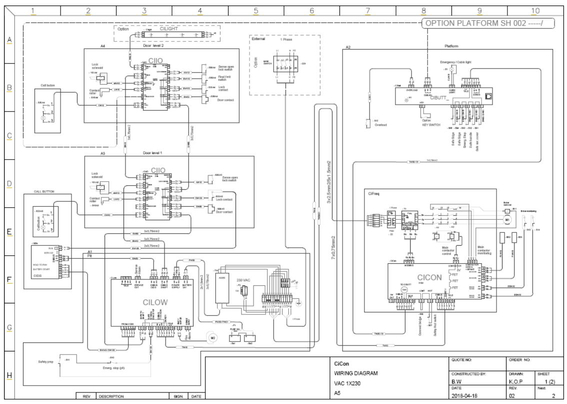


Figure 4: Parts list connection diagram (A5-1X230 Part List).

manuals, and maintenance instructions (Cibes Lift, 2017; Sheridan Lifts, 2018; Kone, 2020).

Lift documents cover design and function instructions; operating instructions, including normal and emergency situations; maintenance instructions as well as safety instructions; standards and directives; parts' lists and descriptions, and a number of diagrams.

As an example of a platform-lift operating instruction, Fig. 2 is a diagram of the control panel within a lift car as well as a description table taken from the documentation (Cibes Lift, 2017). Concepts for the domain can be acquired from this diagram, since the control panel itself would become the class `Control_Panel` of the SLO ontology.

This type of diagrams is helpful to map out concepts of the domain, but within these documents, these diagrams are only presented for larger parts of the system and are not available for the smaller components. So, other information are required for a deeper capture of concepts. Thus, along with diagrams such as shown in Fig. 2, these documents also include lists of parts, as displayed in Fig. 3. An example of this could be that contents of the 'Description' column of the table shown in Fig. 3 can be mapped into SLO classes, while the contents of 'Manufacturer/Supplier' and 'Part no' columns can be used when gauging individuals for the SLO concepts.

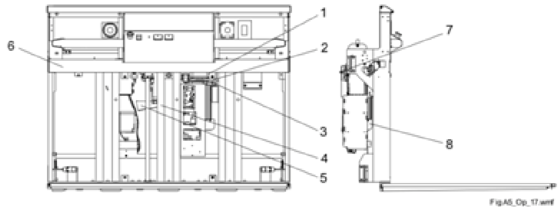
While Fig. 2 and 3 are useful for laying the foundations of the SLO ontology domain and can help in capturing initial main concepts, diagrams such as Fig. 4 help to establish relationships between these concepts.

Other examples of how concepts can be extracted from documents are provided in Figs. 5-6.

Figure 5 is a diagram of the machine area behind the lift car's service cover from the Cibes A5000 operation manual (OM) (Cibes Lift, 2017). This diagram allowed to set concepts for the machine area of the platform-lift system.

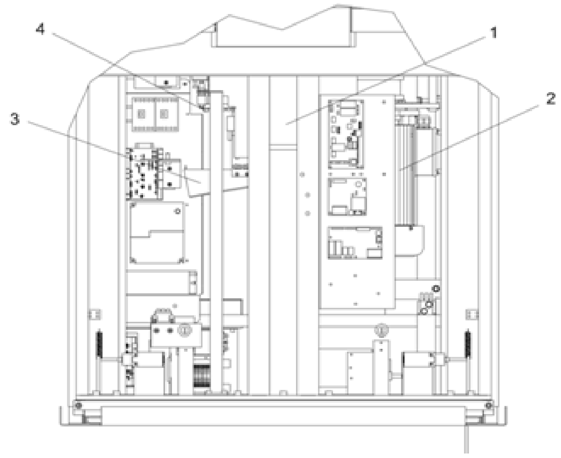
Figure 6 is a diagram showcasing the components behind the service cover of the platform-lift system (Cibes Lift, 2017). This diagram helped in capturing some concepts of the lift maintenance service, which are of great importance due to the safety consequences if any classes or relationships are missed out or mapped out incorrectly.

Hence, SLO domain has been built following a middle-out approach. Indeed, documents, such as the ones illustrated in Figs. 2-6, have been an aid in apprehending the SLO ontology domain. Once a set of concepts and relationships has been extracted from these documents, it opened the way for capturing further the domain by adding data and object properties as well as establishing relationships with every concept within the domain. After the initial ontology domain was established, it was about repeatedly going through the process again and discussing with the stakeholders such as domain experts, mechanical engineers, electrical engineers, product designers, computer scientists, ontologists, industrial partners, manufacturers, vendors, service providers, users, etc. to gather any additional concepts that should be added or removed, whether that be through discussions or further documents.



Item	Description	Item	Description
1	Drive nut	5	Oil container
2	Motor, fixing screw	6	Brake disc
3	Lifting screw	7	Adjustment screw, belt tension
4	Lubricating sponge	8	Motor

Figure 5: Diagram of the machine area of the platform-lift system (A5000 OM Manual).



Item	Description	Item	Description
1	Lifting nuts (hidden)	3	Oil container
2	Motor	4	Adjustment screws

Figure 6: Diagram of the components behind the service cover of the platform-lift system (A5000 OM Manual).

2.2.2 CONCEPT CODING

The knowledge coding has been done in Descriptive Logic (DL) (Black et al., 2021) and uses temporal-interval logic relations as introduced in (Olszewska, 2016).

As an example of concept formalization for the platform-lift design of the lift car's control panel concept, which has been described in Section 2.1, the class of `Control_Panel` is defined in DL, as follows:

$$
\begin{aligned}
Control\_Panel \sqsubseteq\ & Lift\_Car \\
& \sqcap \exists hasPart_{=Alarm\_Button} \\
& \sqcap \exists hasPart_{=Control\_Button} \\
& \sqcap \exists hasPart_{=Emergency\_Stop} \\
& \sqcap \exists hasPart_{=Service\_Cover\_Lock}.
\end{aligned} \tag{1}
$$

As another example of concept formalization, the platform-lift service consisting in the machine area's maintenance, which has been mentioned in Section 2.1, can be formulated in temporal DL, as follows:

$$
\begin{aligned}
Machine\_Area\_Maintenance \sqsubseteq\ & Lift\_Maintenance \\
& \sqcap (\diamond t_1)(\diamond t_2) \\
& (MA_1 < MA_2) \\
& \cdot (MA_1 @ t_1 \sqcap MA_2 @ t_2),
\end{aligned} \tag{2}
$$

with MA_1, the maintenance activity defined as 'Oil Container Refill', MA_2, the maintenance activity consisting in 'Lifting Nut Visual Check', and *before*, the temporal-interval relations as defined respectively in temporal DL:

$$
\begin{aligned}
P_i < P_j \equiv before(P_i @ t_i, P_j @ t_j) \sqsubseteq\ & Temporal\_Relation \\
& \sqcap (\diamond t_i)(\diamond t_j) \\
& (t_{i^+} < t_{j^-}) \\
& \cdot (P_i @ t_i \sqcap P_j @ t_j),
\end{aligned} \tag{3}
$$

where the temporal DL symbol $\diamond$ represents the temporal existential qualifier, and where a time interval is an ordered set of points $T = \{t\}$ defined by end-points $t-$ and t^+, such as $(t^-, t^+) : (\forall t \in T)(t > t^-) \wedge (t < t^+)$.

2.2.3 CONCEPT INTEGRATION

The integration of the SLO ontology was done using the Web Ontology Language (OWL) and carried out within the Protege software environment v.5.5.0 running HermiT v1.4.3.456 reasoner (Glimm et al., 2014), in order to share the SLO ontology knowledge among stakeholders as well as intelligent agents. Indeed, Protege is a widely-used, open-source ontological environment which has a vast and operating community, exceeding 70,000 users (Rubin et al., 2007), and which is adopted for most of the recent engineering-based ontologies for I4.0 (Sampath Kumar et al., 2019; Tsalapati et al., 2021).

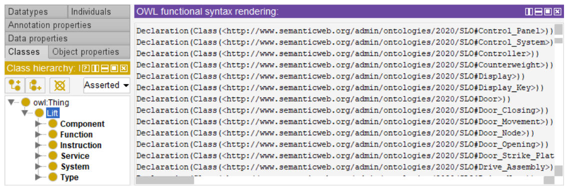


Figure 7: Excerpt of the 'Control_Panel' concept integration within the SLO ontology.

Figure 7 shows a view of Protege integration of some of the smart lift ontology main classes, along with a related OWL/XML code excerpt. Indeed, Protege can generate OWL files that can be accessed from different programming language platforms such as XML. Producing these OWL files that are readable with XML are the final part of the integration process. Transferring the OWL files to XML format allows for a broad range of systems the do;qin ontology could then fully operate on.

It is worth noting that SLO version v2.0.0 contains 749 axioms and 144 classes. Moreover, SLO v2.0.0 includes 16 object properties and 21 data properties. As an example, the SLO ontology defines the class `Control_Panel`, its relationships such as `hasPart` and individuals (e.g., `CibesA5ControlPanel`). In particular, `hasPart` is an object property which domain is 'System' concept and the range is 'Component' concept. The object property `hasPart` is transitive and has an inverse property called `isPartOf`. On the other hand, properties involving lift's key parameters, such as rated speed and rated load, or numeral properties such as the part number have been set as data properties. Further evaluation of these classes, object properties, and data properties is provided in Section 3.1.

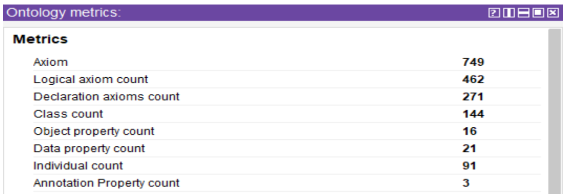
3 VALIDATION AND DISCUSSION

The developed SLO ontology has been evaluated both quantitatively and qualitatively in a series of experiments as described in Sections 3.1, while its documentation is mentioned in 3.2.

3.1 ONTOLOGY EVALUATION

Ontology evaluation is concerned mostly with two chief factors, namely, quality and correctness (Hlomani and Stacey, 2014).

SLO quality evaluation used metrics such as presented in (Tartir et al., 2018). The computed values by Protege are presented in Fig. 8.


Ontology metrics:

Metrics	
Axiom	749
Logical axiom count	462
Declaration axioms count	271
Class count	144
Object property count	16
Data property count	21
Individual count	91
Annotation Property count	3

Figure 8: Main metric values of the SLO ontology.

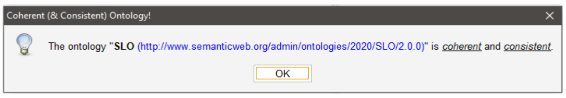


Figure 9: Result of the OntoDebugger when run on SLO ontology.

It is worth noting that, in practice, a trade-off should be achieved between computational efficiency and completeness. Actually, SLO contains so far 749 axioms and 144 classes, while it is processed by HermiT in 2766ms and then performs DL queries in real time.

Moreover, SLO cohesion could be assessed using the number of root classes which is equal to 1, the number of leaf classes which is equal to 112, and the average depth which is equal to 3. All these metrics indicate SLO shows promising performance for real-world deployment.

Besides, the Protege OntoDebugger v0.2.2 allows to automatically check the ontology consistency and coherence. The result of this check for our SLO ontology is successful, as illustrated in Fig. 9.

On the other hand, in Protege, DL Query v4.0.1 allows for an evaluation to be carried out where the two factors of quality and correctness are closely evaluated and achieved. In particular, the ontology correctness could be assessed through experiments running DL queries based on competency questions (Gruninger, 1995).

A first test scenario addresses the competency questions of the type: 'What components are part of the platform-lift control panel?'. For this purpose, we test the object property `hasPart` called on the class `Control_Button` through the DL query: `hasPart some Control_Button`, as in Eq. (1), and the correct answer is provided by our SLO system in Fig. 10(a).

A second test scenario tries to answer the competency question: 'What are the components of the platform-lift Cibes A5 control system?'. A query involving the object property `isPartOf` can be called an instance of the class `Control_System`. The related instances of the 5 modules of the control system are correct, as illustrated in Fig. 10(b).

A third test scenario covers competency questions such as 'Who is the supplier of the Voice System

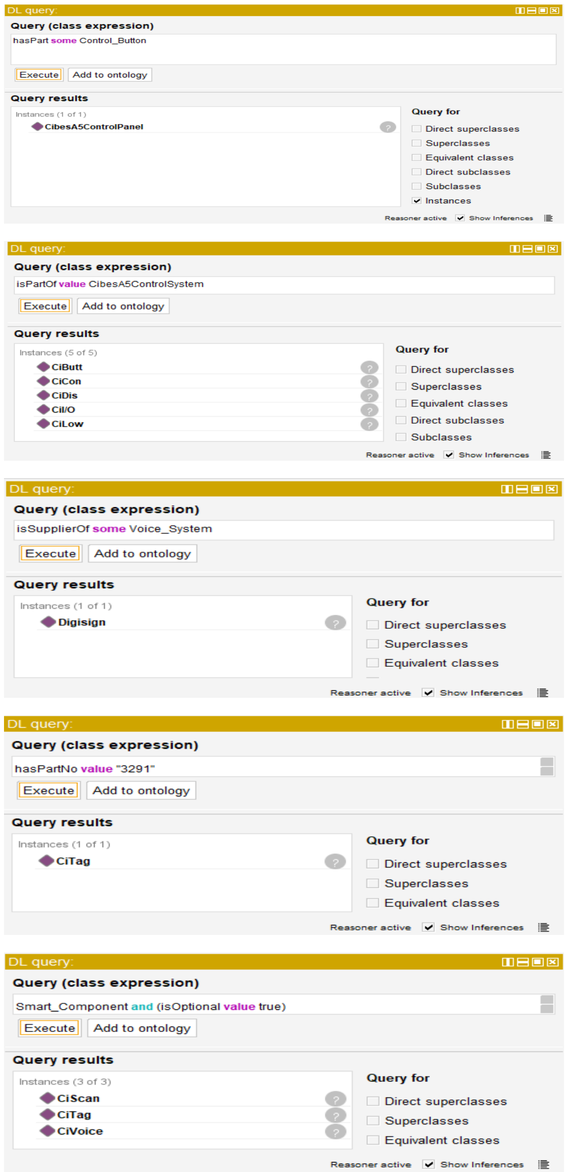


Figure 10: Some samples of query results in relation to the competency questions.

part?'. Hence, the object property `isSupplierOf` is used in the DL Query, as follows: `isSupplierOf some Voice_System`. The corresponding supplier is successfully found in Fig. 10(c).

On the other hand, few experiments have tested data properties such as `hasPartNo` to respond to the competency question: 'What component corresponds to the part number 3291?'. So, we run the DL query: `hasPartNo value ''3291''`, and the component name is correctly displayed on Fig. 10(d).

A further experiment focused on competency questions such as 'Is the fingerprint reader optional?', and thus involved the data property `isOptional`. An example of DL query is: `Smart_Component and (isOptional value true)`, and the results successfully provided by SLO are shown in Fig. 10(e).

In all these experiments targeting classes, individuals, object properties, and data properties, SLO ontology provided 100% correct answers, and no inconsistency has been observed.

3.2 ONTOLOGY DOCUMENTATION

The SLO ontology has been documented and evaluated, as reported in Section 2 and 3.1, respectively. To recap, SLO is a middle-out, domain ontology which has been collaboratively built using Enterprise Ontology methodology. Hence, SLO domain knowledge is based on non-ontological resources such as primary sources, e.g. lift documentation, operational manuals, safety standards, etc., and has been elicited through collaboration with lift domain experts, including lift designers, mechanical and electrical engineers, as well as elevator industry partners. Moreover, SLO ontology has been iteratively developed, with its first version defining 476 axioms and its current, second version containing 749 axioms.

The SLO ontology has not reused any existing ontology, since it is the first ontology in its kind for the smart lift domain. Indeed, some attempts have been made in the past to develop expert systems (Marcus et al., 1987) and knowledge-based systems (Corsar and Sleeman, 2007) for rudimentary elevators, but, on one hand, these works had a limited scope, being focused on the sole design aspect and not embracing all the lift's modern services and, on the other hand, they contained only very few components and parameters, not representing the current, complex smart lift domain.

It is worth noting that SLO domain ontology covers all the smart lift services, addresses the cutting-edge, smart lift domain, and also lays down the foundation for smart lift's digital-twin modeling. Moreover, SLO domain ontology could be used in conjunction with other I4.0 ontologies such as ontologies for IoT (Ma et al., 2014) or other robotics and automation ontologies (Fiorini et al., 2017) for further integration in smart environments.

4 CONCLUSIONS

Since I4.0 has the ability to create new business capabilities and service opportunities, while requiring interoperable technologies, this work is focused on the development of an ontology for the smart lift application, in collaboration with the elevator industry. Our ontology aims to formalize smart lift services, such as smart lift design, operation, and maintenance, e.g., leading to lift automated design for mass customization as well as multimodal operation and AI-enhanced maintenance. Hence, the proposed smart lift ontology (SLO) has the potential to provide the elevator industry with I4.0 benefits, contributing toward innovative smart products, smart machines, and augmented operators, suitable for real-world deployment in context of smart cities and smart factories.

Acknowledgment

The authors would like to thank Innovate UK and Consult Lift Services Ltd for the support of this work.

REFERENCES

Al Sunny, S. M. N., Liu, X. F., and Shahriar, M. R. (2017). MtComm: A semantic ontology based internet scale communication method of manufacturing services in a cyber-physical manufacturing cloud. In *Proceedings of the IEEE International Congress on Internet of Things*, pages 121–128.

Albani, A. and Dietz, J. (2007). Benefits of enterprise ontology for the development of ICT-based value networks. In *Software and Data Technologies*, pages 3–22. Springer.

Alcacer, V. and Cruz-Machado, V. (2019). Scanning the Industry 4.0: A literature review on technologies for manufacturing systems. *Engineering Science and Technology*, 22(3):899–919.

Berger, R. (2020). Emerging technologies could lift elevator market.

Bertolazzi, P., Krusich, C., Missikoff, M., and Manzoni, V. (2001). An approach to the definition of a core enterprise ontology: CEO. In *Proceedings of the International Workshop on Open Enterprise Solutions: Systems, Experiences, and Organizations*, pages 14–15.

Black, R., Davenport, J. H., Olszewska, J. I., Roessler, J., Smith, A. L., and Wright, J. (2021). *Artificial Intelligence and Software Testing: A Practical Guide to Quality*. BCS Press.

Bonci, A., Pirani, M., and Longhi, S. (2017). Robotics 4.0: Performance improvement made easy. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8.

Brings, J., Daun, M., Hildebrandt, C., and Torsleff, S. (2018). An ontological context modeling framework for coping with the dynamic contexts of cyber-physical systems. In *Proceedings of the International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*, pages 396–403.

Burns, M., Griffor, E., Balduccini, M., Vishik, C., Huth, M., and Wollman, D. (2018). Reasoning about smart city. In *Proceedings of the IEEE International Conference on Smart Computing*, pages 381–386.

Canito, A., Aleid, K., Praca, I., Corchado, J., and Marreiros, G. (2020). An ontology to promote interoperability between cyber-physical security systems in critical infrastructures. In *Proceedings of the IEEE International Conference on Computer and Communications (ICCC)*, pages 553–560.

Cecil, J., Albuhamood, S., and Cecil-Xavier, A. (2018). An Industry 4.0 cyber-physical framework for micro devices assembly. In *Proceedings of the IEEE International Conference on Automation Science and Engineering (CASE)*, pages 427–432.

Cheng, H., Zeng, P., Xue, L., Shi, Z., Wang, P., and Yu, H. (2016). Manufacturing ontology development based on Industry 4.0 demonstration production line. In *Proceedings of the IEEE International Conference on Trustworthy Systems and their Applications*, pages 42–47.

Cibes Lift (2017). Operating and Maintenance Instructions. J45017.

Cockburn, I. M., Henderson, R., and Stern, S. (2018). The impact of artificial intelligence on innovation. No. w24449.

Corsar, D. and Sleeman, D. (2007). KBS development through ontology mapping and ontology driven acquisition. In *Proceedings of the ACM International Conference of Knowledge Capture (K-CAP)*, pages 23–30.

Costanzo, A., Faro, A., Giordano, D., and Spampinato, C. (2016). Implementing cyber physical social systems for smart cities: A semantic web perspective. In *Proceedings of the IEEE Annual Consumer Communications and Networking Conference*, pages 1–2.

Derler, P., Lee, E. A., and Sangiovanni Vincentelli, A. (2011). Modeling cyber-physical systems. *Proceedings of the IEEE*, 100(1):13–28.

Dietz, J. and Mulder, H. (2020). *Enterprise Ontology*. Springer.

Dopico, M., Gomez, A., De La Fuente, D., Garcia, N., Rosillo, R., and Puche, J. (2016). A vision of Industry 4.0 from an artificial intelligence point of view. In *Proceedings of the IEEE International Conference on SArtificial Intelligence*, pages 407–413.

Engel, G., Greiner, T., and Seifert, S. (2018). Ontology-assisted engineering of cyber-physical production systems in the field of process technology. *IEEE Transactions on Industrial Informatics*, 14(6):2792–2802.

Feki, M. A., Kawsar, F., Boussard, M., and Trappeniers, L. (2013). The Internet of Things: The next technological revolution. *IEEE Computer*, 46(2):24–25.

Fernandez, M., Gomez-Perez, A., and Juristo, N. (1997). Methontology: From ontological art towards ontological engineering. In *Proceedings of the AAAI Spring Symposium Series on Ontological Engineering*, pages 33–40.

Fernandez-Lopez, M. and Gomez-Perez, A. (2002). Overview and analysis of methodologies for building ontologies. *Knowledge Engineering Review*, 17(2):129–156.

Ferrer, B. R., Mohammed, W. M., Martinez Lastra, J. L., and Strzelczak, S. (2019). A semantic workbench for editing, querying, navigating and distributing ontologies for cognitive manufacturing. In *Proceedings of the IEEE Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pages 2767–2772.

Fiorini, S. R., Bermejo-Alonso, J., Goncalves, P., Pignaton de Freitas, E., Olivares Alarcos, A., Olszewska, J. I., Prestes, E., Schlenoff, C., Ragavan, S. V., Redfield, S., Spencer, B., and Li, H. (2017). A suite of ontologies for robotics and automation. *IEEE Robotics and Automation Magazine*, 24(1):8–11.

Fox, M. S. and Gruninger, M. (1998). Enterprise modeling. *AI Magazine*, 19(3):109–122.

Glimm, B., Horrocks, I., Motik, B., Stoilos, G., and Wang, Z. (2014). HermiT: An OWL 2 reasoner. *Journal of Automated Reasoning*, 53(3):245–269.

Gomez-Perez, A., Fernandez-Lopez, M., and Corcho, O. (2004). *Ontological Engineering*. Springer-Verlag.

Gorecky, D., Schmitt, M., Loskyll, M., and Zuhlke, D. (2014). Human-machine-interaction in the Industry 4.0 era. In *Proceedings of the IEEE International Conference on Industrial Informatics (INDIN)*, pages 289–294.

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal Human-Computer Studies*, 43(5-6):907–928.

Gruninger, M. and Fox, M. S. (1995). The Role of Competency Questions in Enterprise Engineering. In *Benchmarking - Theory and Practice*, pages 22–31. Springer.

Gruninger, M.and Fox, M. (1995). Methodologies for the design and evaluation of ontologies. In *Proceedings of the IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing*, pages 6.1–6.10.

Guo, Y., Qasem, A., Pan, Z., and Heflin, J. (2007). A requirement driven framework for benchmarking semantic web knowledge base systems. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):297–309.

Gyrard, A., Patel, P., Sheth, A., and Serrano, M. (2016). Building the web of knowledge with smart IoT applications. *IEEE Intelligent Systems*, 31(5):83–88.

Hildebrandt, C., Kocher, A., Kustner, C., Lopez-Enriquez, C.-M., Muller, A. W., Caesar, B., Gundlach, C. S., and Fay, A. (2020). Ontology building for cyber-physical systems: Application in the manufacturing domain. *IEEE Transactions on Automation Science and Engineering*, 17(3):1266–1282.

Hlomani, H. and Stacey, D. (2014). Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey. *Semantic Web Journal*, 1(5):1–11.

Hoon, C. S. (2006). *Microcontroller based lift control system*. PhD thesis.

Hoyes, C. and Mair, R. (2020). Game changers in the elevator market.

Ide, N. and Pustejovsky, J. (2010). What does interoperability mean, anyway? Toward an operational definition of interoperability for language technology. In *Proceedings of the International Conference on Global Interoperability for Language Resources*, pages 12–8.

Jones, D., Bench-Capon, T., and Visser, P. (1998). Methodologies for ontology development. In *Proceedings of the IFIP World Computer Congress*, pages 62–75.

Jost, J., Kirks, T., and Mattig, B. (2017). Multi-agent systems for decentralized control and adaptive interaction between humans and machines for industrial environments. In *Proceedings of the IEEE International Conference on System Engineering and Technology*, pages 95–100.

Kalyazina, D. and Kashevnik, A. (2018). Socio-cyberphysical system resource semantic interoperability: General scenarios and ontology. In *Proceedings of the Conference of Open Innovations Association*, pages 320–326.

Kannengiesser, U. and Muller, H. (2013). Towards agent-based smart factories: A subject-oriented modeling approach. In *Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, pages 83–86.

Koh, L., Orzes, G., and Jia, F. J. (2019). The fourth industrial revolution (Industry 4.0): Technologies disruption on operations and supply chain management. *International Journal of Operations and Production Management*, 39(6-8):817–828.

Kone (2020). Lifting the elevator industry to a new era.

Kunold, I., Wohrle, H., Kuller, M., Karaoglan, N., Kohlmorgen, F., and Bauer, J. (2019). Semantic interoperability in cyber-physical systems. In *Proceedings of the IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pages 797–801.

Lee, E. (2019). A fundamental look at models and intelligence. In *Proceedings of the IEEE/ACM International Workshop on Software Engineering for Smart Cyber-Physical Systems*, page 1.

Lewandowski, R. and Olszewska, J. I. (2020). Automated task scheduling for automotive industry. In *Proceedings of the IEEE International Conference on Intelligent Engineering Systems*, pages 159–164.

Ma, M., Wang, P., and Chu, C.-H. (2014). Ontology-based semantic modeling and evaluation for Internet of Things applications. In *Proceedings of the IEEE International Conference on Green Computing and Communications*, pages 24–30.

Marcus, S., Stout, J., and McDermott, J. (1987). VT: An expert elevator designer that uses knpwledge-based backtracking. *AI Magazine*, 8(4):41–58.

Marr, B. (2018). What is Industry 4.0? Here's a super easy explanation for anyone.

Nilsson, J. and Sandin, F. (2018). Semantic interoperability in Industry 4.0: Survey of recent developments and outlook. In *Proceedings of the IEEE International Conference on Industrial Informatics (INDIN)*, pages 127–132.

Nott, G. (2018). IoT, cloud and machine learning giving elevator giants a lift.

Olszewska, J. I. (2016). Temporal interval modeling for UML activity diagrams. In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development (KEOD)*, pages 199–203.

Olszewska, J. I. (2020). AI-T: Software testing ontology for AI-based systems. In *Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (KEOD)*, pages 291–298.

Olszewska, J. I. (2021). Ontologies for Software Testing. In *Artificial Intelligence and Software Testing: A Practical Guide to Quality*, pages 68–77. BCS Press.

Olszewska, J. I. and Allison, I. K. (2018). ODYSSEY: Software development life cycle ontology. In *Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (KEOD)*, pages 303–311.

Onag, G. (2019). AI and IoT are the keys to smarter lifts and escalators.

Pignaton de Freitas, E., Olszewska, J. I., Carbonera, J. L., Fiorini, S., Khamis, A., Sampath Kumar, V. R., Barreto, M., Prestes, E., Habib, M., Redfield, S., Chibani, A., Goncalves, P., Bermejo-Alonso, J., Sanz, R., Tosello, E., Olivares Alarcos, A., Konzen, A. A., Quintas, J., and Li, H. (2020). Ontological concepts for information sharing in cloud robotics. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–14.

Rubin, D. L., Noy, N. F., and Musen, M. A. (2007). Protege: A tool for managing and using terminology in radiology applications. *Journal of Digital Imaging*, 20(1):34–46.

Sampath Kumar, V. R., Khamis, A., Fiorini, S. R., Carbonera, J. L., Olivares-Alarcos, A., Habib, M., Goncalves, P., Li, H., and Olszewska, J. I. (2019). Ontologies for Industry 4.0. *Knowledge Engineering Review*, 34:1–14.

Sheridan Lifts (2018). The most common lift faults. Sheridan Lifts.

Siikonen, M.-L. (1997). *Elevator group control with artificial intelligence*. PhD thesis.

Smirnov, A., Kashevnik, A., Mikhailov, S., Mironov, M., and Petrov, M. (2016). Ontology-based collaboration in multi-robot system: Approach and case study. In *Proceedings of the IEEE System of Systems Engineering Conference*, pages 1–6.

Steinmetz, C., Rettberg, A., Ribeiro, F. G. C., Schroeder, G., and Pereira, C. E. (2018). Internet of Things ontology for digital twin in cyber physical systems. In *Proceedings of the IEEE Brazilian Symposium on Computing Systems Engineering*, pages 154–159.

Sujith, S., Atif, Y., Sheng, Q. Z., and Maamar, Z. (2011). Web of Things: Description, discovery and integration. In *Proceedings of the IEEE International Conference on Internet of Things*, pages 9–15.

Syamili, C. and Rekha, R. V. (2017). Ontology engineering methodologies: An analytical study.

Tang, H., Li, D., Wang, S., and Dong, Z. (2018). CASOA: An architecture for agent-based manufacturing system in the context of Industry 4.0. *IEEE Access*, 6:12746–12754.

Tartir, S., Arpinar, I. B., Moore, M., Sheth, A. P., and Aleman-Meza, B. (2018). OntoQA: Metric-based ontology quality analysis. In *IEEE International Conference on Data Mining Workshop*, pages 559–564.

Thyssen Krupp (2014). Orion - Gulliver Vertical Platform Lift. Installation Manual. INS-OR04-EN-05.

Torsleff, S., Hildebrandt, C., Daun, M., Brings, J., and Fay, A. (2018). Developing ontologies for the collaboration of cyber-physical systems: Requirements and solution approach. In *Proceedings of the IEEE International Workshop on Emerging Ideas and Trends in the Engineering of Cyber-Physical Systems*, pages 25–32.

Tsalapati, E., Tribe, J., Goodall, P., Young, R. I. M., Jackson, T., and West, A. (2021). Enhancing RFID system configuration through semantic modelling. *Knowledge Engineering Review*, pages 1–30.

Voinov, A. and Senokosov, I. (2021). Ontological models of cyber physical systems. *Journal of Physics: Conference Series*, 1889(2):1–10.

Wan, J., Yin, B., Li, D., Celesti, A., Tao, F., and Hua, Q. (2018). An ontology-based resource reconfiguration method for manufacturing cyber-physical systems. *IEEE/ASME Transactions on Mechatronics*, 23(6):2537–2546.

Wilding, S., Walker, P., Clinton, S., Williams, D., and Olszewska, J. I. (2020). Safe human-computer interface based on an efficient image processing algorithm. In *Proceedings of the IEEE International Symposium on Computational Intelligence and Informatics (CINTI)*, pages 65–70.

Xu, X. and Hua, Q. (2017). Industrial big data analysis in smart factory: Current status and research strategies. *IEEE Access*, 5:17543–17551.

Zhong, R. Y., Xu, X., Klotz, E., and Newman, S. T. (2017). Intelligent manufacturing in the context of Industry 4.0: A review. *Engineering*, 3(5):616–630.

Zhu, W., He, H., and Wang, Z. (2017). Ontology-based mission modeling and analysis for system of systems. In *Proceedings of the IEEE International Conference on Smart Data (SmartData)*, pages 538–544.