Kukkonen, Ville; Kücükavcı, Ali; Seidenschnur, Mikki; Rasmussen, Mads Holten; Smith, Kevin Michael; Hviid, Christian Anker

An ontology to support flow system descriptions from design to operation of buildings

Published in:
Automation in Construction

DOI:
10.1016/j.autcon.2021.104067

Published: 01/02/2022

Document Version
Publisher's PDF, also known as Version of record

Published under the following license:
CC BY

Please cite the original version:
Kukkonen, V., Kücükavcı, A., Seidenschnur, M., Rasmussen, M. H., Smith, K. M., & Hviid, C. A. (2022). An ontology to support flow system descriptions from design to operation of buildings. Automation in Construction, 134, [104067]. https://doi.org/10.1016/j.autcon.2021.104067
An ontology to support flow system descriptions from design to operation of buildings

Ville Kukkonen\textsuperscript{a,b,*,} Ali Kücükavci\textsuperscript{c}, Mikki Seidenschnur\textsuperscript{c,d}, Mads Holten Rasmussen\textsuperscript{e}, Kevin Michael Smith\textsuperscript{c}, Christian Anker Hviid\textsuperscript{c}

\textsuperscript{a} Department of Electrical Engineering and Automation, Aalto University, Espoo, Finland
\textsuperscript{b} Granlund, Helsinki, Finland
\textsuperscript{c} Department of Civil Engineering, Technical University of Denmark, Copenhagen, Denmark
\textsuperscript{d} Ramboll, Copenhagen, Denmark
\textsuperscript{e} NIRAS, Allerød, Denmark

ARTICLE INFO

Keywords:
Building information modeling
HVAC
Semantic web
Ontology
Linked data

ABSTRACT

The interoperability of information from design to operations is an acknowledged challenge in the fields of architecture, engineering and construction (AEC). As a potential solution to the interoperability issues, there has been increasing interest in how linked data and semantic web technologies can be used to establish an extendable data model. Semantic web ontologies have been developed for the AEC domain, but an ontology for describing the energy and mass flow between systems and components is missing. This study proposes the Flow Systems Ontology (FSO) for describing the composition of flow systems, and their mass and energy flows. Two example models are expressed using FSO vocabulary. SPARQL Protocol and RDF Query Language (SPARQL) queries are performed to further demonstrate and validate the ontology. The main contribution consists of developing FSO as an ontology complementary to the existing ontologies. Finally, the paper introduces a roadmap for future developments building on FSO.

1. Introduction

The stakeholders in the architecture, engineering and construction (AEC) industry collaborate on complex, multidisciplinary projects that span years and produce large quantities of information besides the finished physical product. Since the introduction of Building Information Modeling (BIM), the level of information in the models has increased to support even more complex use cases and additional disciplines. While BIM authoring tools widely support the Industry Foundation Classes (IFC) data model, which enables vendor-independent information exchange, it has its challenges in terms of adaptability and extensibility [1]. Further, the information storage and exchange are generally based on static documents, which poses challenges such as propagating changes in dynamically derived information [2].

In addition to the information produced during design and construction, the increasing amount of data collection in building operation and maintenance (O&M) has put pressure on improving data interoperability between AEC stakeholders. Advancements in computing and sensing technology as well as practical access to real-time data of buildings have made data analytics increasingly applicable in the O&M of buildings [3,4]. Recently, BIM models have been used to inform the configuration and deployment of building management services such as automated fault detection and diagnostics [5–7]. Although building automation system (BAS) metadata has been successfully represented in IFC, there are still concerns as to whether IFC is a suitable data model for representing information such as BAS control and communication [8]. Additionally, as many existing buildings have been designed and built without BIM, the deployment of building services requires other, more lightweight formats for representing information acquired from targeted buildings [9]. The lack of data interoperability in BIM has contributed to a performance gap between the constructed building and the BIM model [10].

The World Wide Web Consortium (W3C) with their academic and industrial partners have published a set of standards that support the

https://doi.org/10.1016/j.autcon.2021.104067
Received 18 November 2020; Received in revised form 5 November 2021; Accepted 23 November 2021
Available online 9 December 2021
0926-5805/© 2021 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).
implementations of semantic web technologies, with a vision of “web of data” beyond the “web of documents”.

The technologies include Resource Description Framework (RDF) [11], SPARQL Protocol and RDF Query Language (SPARQL) [12], and Web Ontology Language (OWL) [13]. RDF is used for describing graphs of data as sets of triples consisting of a subject, a predicate, and an object, as illustrated in Fig. 1. SPARQL defines a language for formulating queries against graphs expressed in RDF. Finally, OWL enables the definition of ontologies, which codify vocabularies and support reasoning over RDF data by defining the semantics of those vocabularies.

The AEC industry has seen an increasing interest in semantic web technologies [1]. The technologies have been proposed as a foundation for solving some of the interoperability issues, and for moving from document-based collaboration towards network-based collaboration [1, 2]. Semantic web technologies, in particular ontology development and their applications, have also been actively studied in the field of industrial manufacturing. Applications range from integrating legacy data sources in the design and operation of manufacturing processes [14], to supporting smart Product-Service Systems integrating information from design, manufacturing, and operation of the delivered products [15]. Product-Service Systems are concerned with integrating service offerings with delivered products in order to guarantee operation. While similar information integration requirements and opportunities for new value creation exist in the AEC industry, one challenge is that the information models are rarely focused on the operation phase. Embracing linked data and ontologies is one approach to enable information models that can support both design and operation [15].

Within the AEC industry, the W3C Linked Building Data (LBD) Community Group has communicated the benefits of semantic web technologies and linked data by delivering use cases and best practices to the AEC industry since it was established in 2014. Efforts to improve interoperability using the semantic web technologies include the development of ontologies to describe common semantic vocabularies. To this end, the community group has published ontologies for the AEC industry. Other groups and researchers have also proposed ontologies for various subdomains in different levels of abstraction [7,16-19].

While ontologies for describing certain aspects of the AEC domain have been developed and published, a common standardized ontology to describe the energy and mass flow between the components is missing. Such an ontology is needed to support the linked data descriptions of Heating, Ventilation and Air Conditioning (HVAC) systems. This paper will describe a proposed ontology named Flow Systems Ontology (FSO) to represent the composition of flow systems and their energy and mass flow relationships. The proposed ontology is complementary to the ontologies already proposed by the community. FSO enables lightweight, machine-understandable common descriptions of the HVAC components’ relationships. The vision is to provide a common foundation for describing flow systems in linked data. This would enable extensions that focus on more specific information perspectives for applications such as hydraulic simulation suites, building energy performance simulation (BEPS), building analytics, and diagnostics.

In this paper, two research questions are considered: do the currently existing ontologies support descriptions of energy and mass flow connections between systems and components such as seen in the building systems domain; and if not, could such an ontology be created to support industry use cases in design and operation of buildings? The work has been initiated by industrial partners to address identified gaps in information exchange.

The prominent ontologies related to buildings and their systems are briefly described in Section 2. Following that, the proposed Flow Systems Ontology is described in detail in Section 3, including alignments to some of the existing ontologies. In Section 4, example models of a heating system and an active chilled beam system will be expressed using FSO vocabulary to demonstrate that FSO can describe a given flow system. Additionally, examples of queries enabled by FSO will be demonstrated as a means of validating the ontology and to simulate use cases from both design and operation. In Section 5, the results are discussed and the research is placed in the context of a roadmap of transforming existing information sources to linked data and using that linked data to support information interoperability. Finally, conclusions are presented in Section 6.

2. Background

This section describes a selection of the different ontologies developed for buildings and their systems.

An early effort in bridging the gap between BIM and semantic web technologies is the ifcOWL ontology, which maps the IFC schema to OWL [20]. While ifcOWL is expressed in OWL, it is a very direct translation from the EXPRESS schema of IFC, and is thus not optimized for linked data and semantic web applications [21]. Specifically, it covers multiple domains and uses intermediate aggregation objects which are unidiomatic in RDF.

Building Topology Ontology (BOT) was developed by the W3C Linked Building Data Community Group as a lightweight ontology for describing the connections of zones and elements in a building [21]. BOT constitutes a core vocabulary for supporting multiple subdomains within the AEC industries. It is designed to be extendable to more specific domains as needed. In short, BOT consists of a class taxonomy for zones (building sites, buildings, storeys, and spaces), building elements, which may have different spatial relationships to spaces, and interfaces that qualify connections between zones or between zones and elements.

Brick [22] ontology describes building data points, and it originated as a translation of the Haystack tagging framework to semantic web technologies. While some flow relationships between components are present in Brick, it focuses on classifying the different data points. Because the ontology revolves around data points, the scope of the terminology excludes passive components, such as pipes and ducts. The ontology is not intended to describe entire flow systems, including the distribution systems of ducts and pipes. As such, while it is a potential part of a comprehensive description of an operational building, extensions or other ontologies are required for describing the remaining aspects.

The Smart Energy Aware Systems (SEAS) ontology developed in the EUREKA ITEA 12004 SEAS project consists of multiple modules. At the core are several more abstract ontologies, of which the most relevant in this context is the ontology describing systems and their connections. Specifically, the systems and connections module of SEAS contains classes and relationships to describe virtually isolated systems and their composition and connections. However, the requirement that each system may be a subsystem of at most one supsystem, enforced by making seas:subSystemOf functional, is incompatible with overlapping systems common in HVAC systems. For instance, a heating coil in an air handling unit can be considered a part of both the heating system and the ventilation system, which cannot be expressed with the SEAS systems module.

The latest version 3.1.1 of the Smart Applications REFerence (SAREF) [24] with its extensions for building devices (SAREF4BLDG) version 1.1.2 [25] and system typology patterns (SAREF4SYSST) version 1.1.2 [26] published by the European Telecommunications Standards

Fig. 1. An illustration of the RDF data model of subject, predicate, and object.

1 [https://www.w3.org/standards/semanticweb](https://www.w3.org/standards/semanticweb)
2 [https://www.w3.org/community/lbd](https://www.w3.org/community/lbd)
Institute (ETSI) integrates lessons learned from SEAS. As mentioned previously in this section, the SEAS systems are required to have at most one supersystem. This requirement is not present in SAREF4SYSYS, making it more suitable for describing overlapping systems. However, SAREF4SYSYS is an upper-level ontology defining systems and their connections, and is not specific enough to describe flow systems. SAREF4BLDG, on the other hand, takes building devices from IFC and describes their taxonomy in OWL [25]. SAREF4BLDG constitutes a subset of the domains described in ifcOWL. The translation incorporates the existing device taxonomy from IFC while avoiding the issues of ifcOWL discussed previously in this section. However, SAREF4BLDG does not consider passive components such as pipes and ducts, nor the component connections.

Real Estate Core (REC) is a modular ontology developed to support data integration for smart buildings [27]. It is developed by a consortium that includes major real estate companies. The purpose of REC is to semantically describe a building in the operation phase from the perspective of the building owner.

In summary, previous work includes several ontologies for describing the devices, spaces, and data points in a building. While a fictive Flow Systems Ontology was mentioned in [21], and an ontology module with the same name was submitted as a pull request to SEAS, the work presented here represents a new development from the same ideas. The FSO ontology proposed in this paper is complementary to the outlined ontologies and is not intended to supersede any of them but rather augment some of them. Additionally, SAREF4SYSYS describes systems and their connections, providing a useful foundation or “upper-level” ontology for describing flow relationships in building systems. While FSO is not a direct extension to SAREF, alignments to SAREF4SYSYS and SAREF4BLDG are proposed. Conceptually, FSO could be considered a middle-level ontology between SAREF and more domain-specific ontologies, such as REC.

3. Flow Systems ontology

Flow Systems Ontology is an ontology for describing the energy and mass flow relationships between systems and their components, and the composition of such systems. FSO consists of 14 classes and 23 object properties and has a Description Logic expressivity of ALRI. Instances of fso:System are virtual collections of components, which can be assigned properties such as design requirements, while instances of fso:Component are the tangible objects that are involved in the flow of energy or mass. While this study and the examples focus on building systems, FSO is not intended to be strictly limited to those. While not investigated, it seems feasible that the same abstractions should be applicable in, for example, industrial process systems.

One way to conceptualize a building and its systems is to consider them as two parallel hierarchies with some links between them: one for the spaces, and one for the systems and components. Combining FSO and BOT gives the vocabulary necessary for such a conceptualization, as illustrated in Fig. 2.

The rest of this section provides an overview of the ontology and introduces some of its capabilities. The latest version of the ontology is documented online. First, in Section 3.1, competency questions are enumerated to help describe the scope of the ontology. Next, the ontology terms are described in two subsections: Section 3.2 introduces system composition and component classification, and Section 3.3 introduces the relationships between systems and components. After that, some examples of the reasoning enabled by the ontology are shown in Section 3.4. Finally, proposed alignments to other ontologies are described in Section 3.5.

### 3.1. Competency questions

The competency questions (CQ) for FSO are listed below. Their purpose is to illustrate and define the scope of the ontology by defining questions a model should be able to answer when using FSO.

1. What components does a flow system contain?
2. What subsystems does a flow system contain?
3. Given a flow system, which components are sources or consumers of mass or energy?
4. What components are in the same fluid loop as a given component?
5. What components are up/downstream of a component, and in what order?
6. Given a system with components in a fluid loop, which components are on the supply side and which are on the return side?
7. What kind of component is a given component?

The competency questions will be referred to when discussing the specifics of the ontology. This will be done using a shorthand, such as (CQ1) referring to the first competency question.

### 3.2. System composition and component classification

Fig. 3 shows an overview of the primary terminology of the ontology: fso:Component and fso:System and their relations. Instances of fso:System are virtual collections of components, which may have properties assigned to them. Examples of potential system properties include design specifications, such as supply and return water temperature for a heating system. Systems may have subsystems via fso:hasSubSystem (CQ2), and a system may be a subsystem of more than one supersystem via the inverse property fso:isSubSystemOf. This means that systems may overlap, sharing common subsystems. Sub-properties of the fso:hasSubSystem can be used to describe a more specific composition hierarchy (CQ6):

- fso:hasSupplySystem links a system to its logical “supply” subsystem. For example, a ventilation system could denote all components between the outside and a ventilated zone as a part of its supply system.
- fso:hasReturnSystem links a system to its logical “return” subsystem. For example, a ventilation system could denote all components between the ventilated zone and the outside as a part of its return system.

Additionally, fso:System has more specific subclasses that can be used to denote them being the logical supply and return systems:

- fso:DistributionSystem is the class of systems that are used to distribute mass and/or energy, linking other systems.
- fso:SupplySystem, a subclass of fso:DistributionSystem, is the class of systems that are used to supply mass and/or energy to downstream consumers.
- fso:ReturnSystem, a subclass of fso:DistributionSystem, is the class of systems that are used to return mass and/or energy from downstream consumers.

fso:Component and its subclasses are the tangible objects that participate in the flow of mass or energy. A system may have components via the fso:hasComponent property (CQ1), which has the inverse property fso:isComponentOf. The subclasses of fso:Component are based on the IFC taxonomy of IfcDistributionFlowElement, and represent high-level component types applicable

---

1. https://github.com/thesmartenergy/seas/pull/21
2. https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2/HTML/link/ifcdistributionflowelement.htm
These abstract classifications of components together with the flow relationships enable a variety of use cases, which will be demonstrated in Section 4. If gaps in the classifications are identified by future use cases, the component classes can be revised to account for those.

Components can be asserted to be sources or consumers of systems, denoting the relative directions of energy or mass flows (CQ7). Such relationships can be useful for use cases that involve heating or cooling demand calculations. This is achieved with the following properties:

- `fso:hasSourceComponent` links a system to a component that acts as the source of energy or matter into that system. For example, a heating system may contain a heat exchanger that is connected to a district heating loop as an energy source.
- `fso:hasConsumerComponent` links a system to a component that acts as a consumer of energy or matter from that system. For example, a heating system would have radiators that consume energy by transferring heat to the indoor air of a room.

It is worth noting that a component being a source or a consumer is not an intrinsic property of the component, but rather a relation to a system. For example, a heating coil in a ventilation system will be a source of heat for the ventilation system, but a consumer of heat for the heating system. Similarly, a fan in the ventilation system could be considered a consumer of the electrical network, while not having an assigned source or consumer role in the context of the ventilation system.

### 3.3. Flow relationships

Besides system composition, FSO introduces a vocabulary to describe the flow of energy and matter between the systems and components. A generic `fso:connectedWith` property connects components or systems, and the subproperties can be used to denote more specific relationships in terms of energy and mass flows. An overview of the property hierarchy under `fso:connectedWith` is shown in Fig. 4. The first two layers, or all properties ending in `With`, are symmetrical, while their subproperties ending in `To` or `From` are not.

In particular, the `fso:exchangesFluidWith` subproperty of `fso:connectedWith` has two layers of specificity. The property `fso:feedsFluidTo` and its inverse `fso:hasFluidFedBy` denote the flow order of fluid in a system. `fso:suppliesFluidTo` and `fso:returnsFluidTo` denote the logical supply and return of fluid from a component or system to another (CQ5 & CQ6), and are subproperties of `fso:feedsFluidTo`. These properties can be used to break down a fluid loop of components and systems feeding to another to two sides: a supply and a return side. Note, however, that a `fso:suppliesFluidTo` property path from system A to B to C does not necessarily mean that it is the same fluid flowing through A and C. For systems like water to water heat exchangers, it might be intuitive to denote that the system is being supplied from one side while it itself supplies another.
side, even though in reality the fluid is in two separate loops and hence completely separated. However, in order to be able to follow individual fluid loops (CQ4), such systems should be represented as multiple components. Continuing with the water to water heat exchanger example, a heat exchanger has a primary and a secondary side. FSO includes reasoning mechanisms that help inferring the implicit knowledge in these situations. This will be further elaborated in Section 3.4.

Using only fso:feedsFluidTo in SPARQL property path expressions with the zero-or-more or one-or-more syntax has the problem that it goes through the whole loop, as illustrated in Fig. 5. For example, in a system with a heat exchanger feeding fluid to two radiators that feed fluid back to the heat exchanger, going either forward or backward through fso:feedsFluidTo will connect the radiators. FSO introduces the more specific properties fso:suppliesFluidTo and fso:returnsFluidTo to break up the loops and accommodate this problem.

The heat transfer connections have a hierarchy as well. fso:exchangesHeatWith is a subproperty of fso:connectedWith, and in turn has two directional subproperties that are inverses of each other: fso:transfersHeatTo and fso:transfersHeatFrom. While the property fso:exchangesHeatWith is symmetric, the subproperties are not.

Finally, the ontology defines a third kind of connectivity, fso:exchangesElectricChargeWith. While its subproperties have not been detailed at this stage, the property has been included as it is recognized to be similar to, but different from, the two previously defined connections. Like the other direct subproperties of fso:connectedWith, it too is defined as a symmetric property.

3.4. Reasoning examples

In addition to explicit assertions, semantic web technologies enable deductive reasoning to produce new assertions. This section shows some examples of the reasoning that FSO enables.

Besides components having connections to other components, system-level connections can be inferred from the system composition and component connections. The flow relationship properties discussed in Section 3.3 utilize property chain axioms to imply the existence of system-level connections. If there exists a component that is connected to another component with e.g. fso:suppliesFluidTo, and those
components are part of different systems, then those two systems are also connected by fso:suppliesFluidTo, as shown in Fig. 6. This means that if a pipe in a heating system supplies fluid to a heating coil that is also a part of an air handling unit, then the heating system can be inferred to have a fso:suppliesFluidTo connection to the air handling unit. Note, that because a component can be a component of many systems, overlapping systems are inferred to have these relationships. Additionally, these property chain axioms make the properties reflexive for the systems, i.e. the system on both ends of the property chain can be the same, leading to a self-referential object property to be inferred.

As was mentioned in Section 3.3, systems such as water to water heat exchangers can be modeled as two components to allow tracking of specific fluid circuits. Modeling the heat exchanger as a single component and connecting it with fluid supply upstream and downstream would result in the model appearing as if all components upstream of the heat exchanger also supply fluid to the components downstream of the heat exchanger. If the heat exchanger is instead modeled as a system with two components, a primary side and a secondary side with an fso:transfersHeatTo relationship, this confusion can be avoided. An example of this is illustrated in Fig. 7. The connection between the sides is shown as fso:exchangesHeatWith, as it is the symmetrical superproperty of fso:transfersHeatTo, and will be asserted in both directions by the reasoner.

With the model as shown in Fig. 7, the reasoning combined with SPARQL property path queries enable tracking the fluid and heat flows. Finding the components that are in the same fluid circuit as, for example, the component supplying the primary side of the heat exchanger, is achieved by following the path fso:suppliesFluidTo*/fso:returnsFluidTo* The * in a path expression denotes zero or more matches, and the / denotes a sequence path. If a different view of the network is desired, including those on the other side of such thermal connections, the property fso:exchangesHeatWith could be included in the path.

Additionally, with the model as shown in Fig. 7, the reasoner will infer a system-level connection between the heat exchanger system and the systems around it, as discussed earlier in this section. This means that the heat exchanger system can be inferred to e.g. supply fluid to a heating system. Similarly, if the primary side is a component of a district heating system, it is inferred to transfer heat to the heating system.

3.5. Alignments

SAREF4SYST introduces an ontology pattern for modeling systems and their connections. The alignment of FSO to SAREF4SYST is done by asserting the relevant FSO classes and properties as subclasses and subproperties of their more general counterparts in SAREF4SYST, as shown in Table 1. In particular, both fso:System and fso:Component are aligned as subclasses of s4syst:System.

FSO component taxonomy is largely based on the same IFC taxonomy as SAREF4BLDG, so the alignments are straightforward. However, SAREF4BLDG only considers devices and not passive elements, such as segments and fittings. Additionally, SAREF4BLDG devices are specific to buildings, while FSO does not place such requirements. As such, the classes from SAREF4BLDG are subclassed from FSO, as shown in Table 2.

Table 1: Alignments between FSO and SAREF4SYST.

| Class / property | Subclass/subproperty of |
|------------------|-------------------------|
| fso:System       | s4syst:System           |
| fso:Component    | s4syst:System           |
| fso:connectedWith| s4syst:connectedTo       |
| fso:hasSubSystem | s4syst:hasSubSystem      |
| fso:isComponentOf| s4syst:subSystemOf       |
| fso:isComponentOf| s4syst:subSystemOf       |

Fig. 6. Illustration of system connections inferred from component connections.

Fig. 7. Example of a heat exchanger modeled as two separate components exchanging heat with each other.
4. Example models and use case demonstrations

In this section, example models and use cases are presented to demonstrate the use of FSO. The Turtle and SPARQL files of the examples are openly available in Zenodo [28]. The namespaces for prefixes used throughout the paper are shown in Table 3.

As FSO itself only defines a core vocabulary for describing flow systems in terms of abstract component classes and their relationships, the use cases depend on the use of other ontologies. This demonstrates that while FSO itself does not define all the vocabulary for the use cases, semantic models of the flow systems form structural skeletons that can be augmented with further information.

First, Section 4.1 introduces an example model of an active chilled beam system for one room in order to demonstrate overlapping systems. Section 4.2 describes an example model of a radiator heating system for several rooms that is used later in the use case demonstrations in Sections 4.3 and 4.4. Section 4.3 presents use cases related to the design of flow systems, followed by use cases from the operation of such systems in Section 4.4.

4.1. Active chilled beam system

This section introduces an active chilled beam system example model. Active chilled beam systems combine heating and cooling with ventilation. In this example model, the interface between cooling and ventilation in an active chilled beam has been illustrated to demonstrate the expressiveness of FSO. In addition to FSO, BOT is used with the prefix bot:

Fig. 8 illustrates the system in a schematic annotated with FSO terminology. An active chilled beam has a heat exchanger and an air supply, and is modeled as an fso:Component. The heat exchanger is a terminal connected to the cooling system, and the air supply is a terminal connected to the ventilation system. The heat exchanger transfers heat from the room, and the air supply supplies fluid, i.e. air, to the room. The cooling coil in the ventilation supply is modeled as two components with a heat transfer connection: the cooling coil itself is supplied with fluid by the cooling system piping, and the cooling coil “air side” is supplied with fluid by the ventilation system.

![Fig. 8. Illustration of the active chilled beam system example model as a schematic with FSO terms.](image-url)
Fig. 9. Illustration of the heating system example model as a schematic with FSO annotations. While only a subset of the schematic is annotated here, a similar approach has been followed for the rest of the model to express the whole schematic in RDF. The complete model is used in the use case demonstrations.
The system composition is not illustrated in Fig. 8, but is shown in Listing 1. Four instances of fso:System are modeled: one for the ventilation system, one for the cooling system, one for the cooling coil consisting of the liquid and air sides, and one for the active chilled beam consisting of two terminals.

Listing 1. System composition of the example active chilled beam model expressed in Turtle syntax.

```turtle
inst:Cooling Coil-1 a fso:System ;
    fso:hasComponent
    inst:Cooling Coil-1-Liquid Side ,
    inst:Cooling Coil-1-Air Side .

inst:Cooling System-1 a fso:System ;
    fso:hasSubSystem inst:Beam-1 , inst:Cooling Coil-1 ;
    fso:hasComponent
    inst:Heat Pump-1 ,
    inst:Pump-1 ,
    inst:Pump-2 ,
    inst:Regulation Valve-1 ,
    inst:Regulation Valve-2 ,
    inst:DP-Regulation Valve-1 ,
    inst:DP-Regulation Valve-2 ,
    inst:Pipe-1 ,
    # ... pipes and fittings ...
    inst:Finishing-8 .

inst:Ventilation System-1 a fso:System ;
    fso:hasSubSystem inst:Beam-1 , inst:Cooling Coil-1 ;
    fso:hasComponent
    inst:Cooling Coil-1-Liquid Side ,
    inst:Supply Fan-1 ,
    inst:Cooling Coil-1-Air Side ,
    inst:Duct-1 ,
    inst:Return Vent-1 ,
    inst:Duct-2 ,
    inst:Return Fan-1 .
```

4.2. Heating system

This section describes an example model of a heating system used in the use case demonstrations in Sections 4.3 and 4.4. An illustration of the model is shown in Fig. 9.

The example model depicts a simple heating system in a two-storey residential building. There is a heat exchanger (fso:EnergyConversionDevice) supplying heat to six radiators (fso:Terminal) in four rooms (bot:Space). Rooms 1 and 4 have two radiators each, while rooms 2 and 3 have one radiator each. The fluid is circulated by a pump (fso:FlowMovingDevice) through various pipe segments (fso:Segment) connected with various fittings (fso:Fitting) such as tees and elbows. For balancing, the system has multiple manual balancing valves (fso:FlowController).

The fluid flow is described by connecting the components with fso:suppliesFluidTo on the supply side, i.e. before the radiators, and fso:returnsFluidTo on the return side, i.e. after the radiators. Additionally, the rooms are modeled to contain the radiators with bot:containsElement, while simultaneously the radiators are modeled to transfer heat to the rooms with fso:transfersHeatTo.

The composition of the system is not shown in Fig. 9. This is implemented as shown in Listing 2. There is a top-level system inst:HeatingSystem-A with the supply system inst:A-S and return system inst:A-R. The subsystems have the concrete components. The radiators and the heat exchanger are components of both the supply and the return systems.

Listing 2. Composition of the heating system example model expressed in Turtle syntax.

```
# Composition of the heating system.
inst:Heating System-A a fso:System ;
    fso:hasSupplySystem inst:A-S ;
    fso:hasReturnSystem inst:A-R .

# Composition of the heating supply system.
inst:A-S a fso:SupplySystem ;
    fso:hasComponent ... # supply components

# Composition of the heating return system.
inst:A-R a fso:ReturnSystem ;
    fso:hasComponent ... # return components
```

4.3. Design phase

In the design of flow systems in buildings, there are many subsystems to keep track of for the individual engineer. As mentioned in Section 1, there is a lack of data interoperability. This means that the logic of separate flow systems (ventilation, cooling, heating, etc.) is often interpreted differently by specific disciplines (ventilation designer, cooling designer, heating designer), though they are correlated with each other. Such different interpretations create calculation inconsistencies between different model elements. In the following use cases, examples are carried out, to display the potential use for mechanical designers.

4.3.1. Querying for systems and subsystems

HVAC systems are complex in nature. In large projects, BIM models of HVAC systems consist of possibly hundreds of subsystems. This means that it can be a laborious task to validate that the integrity and the logic between subsystems have been maintained in the modeling process.

Based on the heating system introduced in Section 4.2 a query can be formulated to return any fso:System and their supersystems, along with a string representation of the classes. This is shown in Listing 3.

Listing 3. Querying for systems and subsystems.

```
PREFIX fso: <https://w3id.org/fso#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

# Find all instances of type fso:System, their # supersystems, and system types.
SELECT ?system ?superSystem
WHERE {

    ?system a fso:System .
    OPTIONAL { ?system fso:isSubSystemOf ?superSystem . }
    ?system a ?systemType .
    ?systemType rdfs:subClassOf* fso:System .
}
GROUP BY ?system ?superSystem
```

As seen in Table 4, inst:HeatingSystem-A has two subsystems inst:A-S and inst:A-R. This means that for the heating system, there exist two subsystems of the heating system, which is the supply system.
and return system. The result in Table 4 may not seem useful for this use case, since there are only two subsystems. However, large projects could consist of hundreds of subsystems, which means that it could prove useful to generate an overview to ensure modeling integrity.

4.3.2. Querying for components that serve a room

With large BIM models displayed in 3D, it can be hard to generate a visual overview of the integrity of a mechanical system. In some cases, the designer will only want to see specific flow components and their connections. Whilst visualizations are usually available in existing model viewers, the user does not have full control of which methods are used to display the connection of specific components.

Listing 4 matches all components that serve a specific room and belong to a specific system.

**Listing 4.** Querying for components of a specific system supplying a specific room.

Running the query of Listing 4 returned 27 of the 76 components in the heating system example model shown in Fig. 9. 5 of the components are displayed in Table 5. This use case demonstrates a way to query components by their relationships to systems and indirect connections to zones. This could be used to, for example, support customizable filtering options in a model viewer.

4.3.3. Calculating the mass flow rate of a system

Part of the mechanical designer’s responsibility is to size the flow systems. As mentioned in Section 4.3, the methods used to make the calculations often vary with different designers, meaning that there are calculation inconsistencies between different designers. Therefore, the query in Listing 5 demonstrates a way to calculate the flow rate directly in SPARQL. The mass flow rate formula is written as:

\[ \dot{m} = \frac{\Phi}{(c_p \times \Delta T)} \]

where \( \dot{m} \) is the mass flow rate \((\text{kg}/\text{s})\); \( \Phi \) is the heating/cooling demand \((\text{W})\); \( c_p \) is the specific heat capacity \((\text{J}/(\text{kg} \cdot \text{K}))\); \( \Delta T \) is the temperature difference \((\text{K})\).

The query in Listing 5 returns the supply side mass flow rate demand, converted to kg/h, for each HVAC component based on the downstream heating demand.

**Table 5**

Result of running query in Listing 4 on the heating system example model.

| components       |
|------------------|
| inst:HeatExchanger-1 |
| inst:Pipe-1      |
| inst:Pump-1      |
| inst:Fitting-3   |
| inst:Pipe-6      |
| ...              |

**Table 6**

Result of running query in Listing 5 on the heating system example model.

| element          | downstreamHeatDemand | massflow   |
|------------------|-----------------------|------------|
| inst:HeatExchanger-1 | 5900                 | 254.07     |
| inst:Pipe-1       | 5900                  | 254.07     |
| inst:Pump-1       | 5900                  | 254.07     |
| inst:Fitting-3    | 3800                  | 163.64     |
| inst:Pipe-6       | 800                   | 34.45      |
| ...               |                       |            |
at all observable, may be either directly observable and have actual readings available, or they may be indirectly observable and estimated from the available observations with so-called virtual sensors. One obstacle for setting up continuous monitoring of system variables in buildings is the cost of setup, which is in part caused by varying naming schemes of data points [4]. Using a flow model to describe the relationships of components and connecting the available data points to the components could support various monitoring applications. The following examples show simplified queries supporting use cases from building operations and maintenance.

4.4.1. Querying for flow control components in the same loop
Listing 6 shows an example of a query for retrieving all the flow-altering components in the same loop as a chosen radiator. This could be useful for diagnosing anomalous room temperatures, assuming the components have some control values available. It also showcases a useful pattern that can find all the components in the same loop as a given target using the concepts of fluid supply and return with SPARQL property paths.

| ?ctrlComp | ?kind |
|-----------|-------|
| inst:Pump-1 | fso:FlowMovingDevice |
| inst:RegulationValve-1 | fso:FlowController |
| inst:RegulationValve-7 | fso:FlowController |
| inst:RegulationValve-6 | fso:FlowController |

Listing 6. Querying for flow affecting components in the same loop as a given target.

The result of running the query on the heating system are shown in Table 7. The result shows that there is one pump and three valves in the same loop as the given radiator. Each of these devices could have more information attached to them, such as operating manuals or data labels. This kind of query could be particularly useful if a component, such as a radiator, in the loop is not working as expected. This could be caused by another faulty component in the same loop, such as a stuck or fluctuating valve.

4.4.2. Querying the upstream components
When diagnosing faults in flow components, it is often useful to know which components are found upstream of the component with anomalous behavior. For example, if a room is consistently too cold, it would seem obvious to take a look at the radiator in the room. While a single radiator may be faulty, the fault could also lie upstream of the radiator, possibly causing symptoms in other radiators as well.

An example of how the FSO object properties can be used to query for components upstream of a given component is shown in Listing 7. This information could be useful for an automatic diagnosis tool, or a manual diagnosis user interface.

4.4.3. Calculating the average room temperature error by valve
During the life cycle of a heating system, it is usually necessary to balance the pressure of the liquid circulating in the radiators to ensure occupant comfort and to optimize the heating system’s energy consumption. To monitor that the system is working as intended, it can be useful to track the average temperature error of spaces served by a specific balancing valve.

After augmenting the heating system example model with naive representations of room temperature sensor readings and setpoints, querying the average temperature error of rooms grouped by flow controllers in the same loop could be done as shown in Listing 8. While an actual application likely would not store the values as queried in Listing 8, the query could, for example, be modified to retrieve identifiers used as keys for the sensors and setpoints in another storage system more suited for time series data. Additionally, while the example calculates the average in SPARQL, it could be simply retrieving the data point identifiers for an analysis tool to perform more advanced analysis.

| ?ctrlComp | ?avg |
|-----------|------|
| inst:Pump-1 | fso:FlowMovingDevice |
| inst:RegulationValve-1 | fso:FlowController |
| inst:RegulationValve-7 | fso:FlowController |
| inst:RegulationValve-6 | fso:FlowController |

Listing 8. Average room temperature error grouped by valve device.

A similar approach could be used to discover the average temperature error for each heat source in a more complex system with multiple heat sources.
The result of running the query in Listing 8 on the heating system is shown in Table 9. With the example values, some of the valves have an average error of 0 degrees while one has an error of up to 2.0 degrees. This could indicate an imbalance in the heating system.

### 4.4.4. Projecting to a simplified logical topology

It is often useful to consider the logical topology of the components in a flow system, disregarding the actual flow segments and fittings that make up the physical flow connections. This kind of simplified model could also be the best thing available when modeling a building with no BIM model.

One way to project a detailed model to a more simple one is the CONSTRUCT query form which constructs a new graph. These constructed graphs could be used similarly as materialized views in relational databases, where the data is projected into a certain format for specialized queries, simplifying and improving the efficiency of those queries. For example, dropping the segments and fittings could be useful for fault detection, where the focus is on the active components. An example query that constructs a graph of the fluid supply and return connections between components without flow segments and fittings is shown in Listing 9.

### Listing 9. CONSTRUCT query for a simplified graph.

The result of running the query in Listing 9 is shown in Fig. 10 as a visual graph, manually organized for ease of reading. The result shows that the fluid flows from a central heat exchanger through a pump to all the radiators. Additionally, it can be seen that each radiator has a valve on the return side and that the radiators are grouped into sets of three with second valves. Again, further information could be linked to the components, supporting use cases such as fault diagnosis.

### 5. Discussion and future work

The example models and the example queries described in Section 4 illustrate simplified use cases. They are an attempt at capturing the essentials of more complex, real-world use cases without going into detail. The models and queries show that fluid circuits, such as used in heating, can be isolated and analyzed using the component relationships. They also allude to the potential in expressing and tracking the composition of systems, which is useful as the number of systems increases. For actual extended use cases, more specific information about the model would be required, and thus the vocabulary for describing that information would be required as well. Examples of further information include product details, such as pipe materials and dimensions, as well as functional requirements for spaces, such as ventilation rates. For expressing this information, other ontologies alongside FSO would need to be used or possibly created.

Some of the existing ontologies for the AEC industry that FSO implements were discussed in Section 2, and include the likes of BOT [21] and SAREF [24]. In addition to these, other ontologies relevant to the BIM and linked building data context include Ontology for Property Management (OPM) [2] for managing properties and Ontology for Managing Geometry (OMG) [31] for managing geometry descriptions. It is worth noting that while BIM tools generally view elements, such as...
flow components, as 3D objects with particular attributes, in the linked data approach the geometry of an object is an attribute among others.

Other common representations for information similar to covered by FSO and used in the HVAC and industrial process domains include Process Flow Diagrams (PFDs). While PFDs are a primarily graphical representation, FSO is a computer-interpretable vocabulary for representing some of the information that would be required to generate a PFD. Additionally, while there is significant overlap between the information contents, FSO needs to be combined with other ontologies to represent all the information representable in PFDs. Similarly, a PFD will generally not include all information representable by FSO.

5.1. Limitations of the study

One limitation of the research presented is the lack of use of a formal ontology development methodology, which is something that has been called for in relation to previous ontology developments [9]. Although no formal, structured method is used, some of the ontology engineering best practices have been implicitly followed. Firstly, the existing ontologies have been researched and their reuse has been considered, resulting in some alignments already mentioned in Section 3.5. Secondly, there are efforts at validating the ontology, although the extent of validation is still limited, and more structured validation should be carried out in future work. Finally, competency questions are used to explicate the scope of the proposed ontology.

Another limitation is that while expert knowledge has been used in developing the scope and terms of the ontology, the number of participants is limited mainly to the authors. Besides the component classes based on IFC, the authors’ own experience from different fields has been the primary source of definitions for the ontology and is something that needs to be opened to further refinement by additional domain experts.

Additionally, while the ontology is theorized to support descriptions outside the HVAC domain, these were not investigated. This means that there are likely unknown limitations when applying to other flow system domains.

Further, while FSO is intended to support various applications, including simulations and monitoring, this paper does not use real data from a particular tool. Once real data is included from a tool such as an HVAC simulation, then the paper needs to go into specific details of that application domain, ruling out other domains. Demonstrating the use of FSO in integrating real-world applications would require further – potentially application-specific – ontology and software tooling development. The need to develop new tooling and combine FSO with other ontologies for practical use cases is discussed in Section 5.2 and illustrated in Fig. 11. The proposed ontology only contains the terminology for describing the topology of a flow system in terms of (sub-)system composition, component classifications, and flow connections. It, by design, does not contain the terminology for describing the properties of components or the instrumentation of HVAC processes. As such, FSO alone is not sufficient for describing the inputs or outputs of simulation tools, nor the instrumentation of building automation systems. The use case descriptions presented in Section 4 use imaginary complementary ontologies and data to showcase the potential for integrations on a
conceptual level.

Finally, as briefly mentioned above, it is worth highlighting that the validation has been mostly done using two example models with manually written triples, described in Section 4. To more extensively evaluate the expressiveness, computational characteristics, and potential shortcomings of the ontology, larger models from actual design processes should be used. This would require the development of a tool that is capable of converting a format such as IFC into FSO annotated linked data.

5.2. Roadmap for further development

The development of FSO is envisioned as a part of a three-stage process, pointing to future applications of FSO and further research in supporting information interoperability for flow systems. The stages are illustrated in Fig. 11 and described in more detail next.

Stage 1 is the development of an abstract ontology for describing flow systems, as done in this article. The concrete use of FSO in future applications would be to annotate information either directly, or via more domain-specific ontologies extending FSO, thus enabling reasoning and queries as illustrated in this study. The current version of FSO provides a stepping stone on the way to utilizing semantic web technologies in the context of flow systems such as in HVAC systems, which supports improving data interoperability [1]. The abstract definitions leave room for extending into concrete use cases, allowing alignment via a common upper terminology. The development of the ontology is the first stage in the process as it includes implicit considerations for the use cases downstream and the information available in various sources common in the industry. However, future developments in either direction will inform the refinement of the ontology itself.

Stage 2 consists of the development of tools to convert IFC into linked data representations annotated with FSO. Tools for converting IFC to RDF with IFCOWL exist [29], as well as tools built on top of it for converting to the ontologies developed by the W3C LBD Community Group [30]. This stage is important for enabling the utilization of BIM models created with current modeling practices. While linked data representations of the information are unlikely to replace IFC files in the industry, tools for consistently and transparently breaking the information out of the IFC document context have value for downstream use of the information. Therefore, the primary avenue of research should be integrating IFC into the pipeline of information flow to future applications that use FSO.

Stage 3 involves the development of more specific ontologies extending FSO, and tools such as parsers bridging the gap from linked data representations to existing and new industry applications. Some of the potential applications that either require or could benefit from the knowledge of flow system relationships include: indoor climate simulations; numerical flow analysis; automated fault detection and diagnosis; and sizing of Mechanical, Electrical and Plumbing (MEP) systems. The development of these use cases can be used to further inform the development of FSO, creating a feedback loop for future iterations. In particular, the authors are working to apply FSO as a part of a new hydraulic simulation process, and to provide a data model for fault detection in HVAC systems.

6. Conclusions

Linked data and semantic web technologies have been gaining interest within the architecture, engineering and construction (AEC) industry, which has brought about the development of multiple ontologies within the domain. This study summarized the prominent ontologies in the AEC field and showed that there is no ontology for the description of the mass and energy flow relationships of systems. To fulfill this gap, this research introduced a new ontology for describing the flow systems in terms of their composition and mass and energy flow relationships. To this extent, the study enumerated the competency questions driving the
development of the ontology, and related the term definitions to their relevant competency questions. The proposed ontology was demonstrated and validated using example models of a heating system and an active chilled beam system. Further, queries in SPARQL Protocol and RDF Query Language (SPARQL) were formulated and evaluated against the heating system, simulating various use cases. The ontology development was justified in the context of the bigger picture of information interoperability, discussing the plans for future extensions and information source integrations.

Declaration of Competing Interest

The authors recognize that there is a potential for conflicts of interest via industry affiliations. Ville Kukkonen is working on a doctoral dissertation in Aalto University while also working in Granlund. Ali Kücükavci is working on a doctoral dissertation in Aalto University while also working in Granlund. Ali Kücükavci is working on a doctoral dissertation in Technical University of Denmark, and was sponsored by Tyrén's A/S during the reported study. Mikki Seidenschchr is working on a doctoral dissertation in Technical University of Denmark, while also working in Ramboll A/S. Mads Holten Rasmussen works in NIRAS A/S.

Acknowledgements

This work was supported by Tyrén's A/S; EU-Interreg ÖKS “Data-driven Energy Management in Public Buildings”; Granlund; the K.V. Lindholm foundation; the Ramboll Foundation; and Innovation Fund Denmark.

References

[1] P. Pauwels, S. Zhang, Y.C. Lee, Semantic web technologies in AEC industry: a literature overview, Autom. constr. 73 (2017) 145–165, https://doi.org/10.1016/j.autcon.2016.10.003.
[2] M.H. Rasmussen, M. Lefrançois, P. Pauwels, C.A. Hviid, J. Karlshøj, Managing information source integrations.
[3] K. Hammar, E.O. Wallin, P. Karlberg, D. Håkansson, Integrating IFC objects and facility management work information using semantic web, Autom. constr. 87 (2018) 169, https://doi.org/10.1016/j.autcon.2018.10.020.
[4] K. Kim, H. Kim, W. Kim, K. Kim, J. Yu, Integration of IFC objects and facility management work information using semantic web, Autom. constr. 87 (2018) 173–187, https://doi.org/10.1016/j.autcon.2017.12.016.
[5] S. Tang, D.R. Sheldon, C.M. Eastman, P. Pishbad-Bozorgi, X. Gao, SEM assisted building automation system information exchange using BACnet and IFC, Autom. constr. 110 (2020) 103049, https://doi.org/10.1016/j.autcon.2019.103049.
[6] G.F. Schneider, G.D. Kontes, H. Qiu, F.J. Silva, M. Bucar, J. Malanik, Z. Schindler, P. Andrópolous, P. de Agustín-Camacho, A. Romero-Amorrortu, G. Grün, Design of knowledge-based systems for automated deployment of building management services, Autom. constr. 119 (2020) 103407, https://doi.org/10.1016/j.autcon.2020.103407.
[7] A. Wagner, M.B. Bonduel, P. Pauwels, R. Uwe, Relating geometry descriptions to its derivatives on the web, University College Dublin, Chania, Crete, 2019, pp. 304-313. https://doi.org/10.53490/ec3.2019.146.