

From obXML to the OP Ontology: Developing a Semantic Model for Occupancy Profile*

Serge Chávez-Feria¹[0000-0002-7454-9202], Giorgos Giannakis², Raúl García-Castro¹[0000-0003-3587-0367], and María Poveda-Villalón¹[0000-0003-3587-0367]

¹ Ontology Engineering Group, Universidad Politécnica de Madrid, Spain
serge.chavez.feria@upm.es, rgarcia@fi.upm.es, mpoveda@fi.upm.es
² Hypertech, Greece
g.giannakis@hypertech.gr

Abstract. Building energy performance simulation is vital for predicting energy performance under specific conditions. Even though there exist several calculation methodologies that can provide precise simulation approximations, the accuracy of these tools is affected by the level of detail of its input data, being occupant behavior the main source of uncertainty in building energy performance results. Hence, initiatives such as the obXML schema have been developed recently to get a deeper understanding and to generate a more elaborated description of these data. Besides, the use of semantic web technologies can open opportunities to link behavioral patterns with other building information and related domains allowing interoperability and heterogeneous data integration. This paper presents the Occupancy Profile ontology, a semantic model based on obXML that allows the representation of occupant behaviors and actions inside building spaces. This paper also provides a catalogue of modelling decisions taken during the development and an example.

Keywords: Ontological Engineering · Occupancy Profile · Semantics.

1 Introduction

The usage of building energy performance simulation has gained significant attention recently that stems from its capability to accurately predict the energy performance of the building sector under specific conditions. Among a wide range of calculation methodologies, the 3D zonal-type simulation approximation is frequently adopted in many envisaged and practical use scenarios, as it manages to strike a balance between accuracy and computational complexity. However, the accuracy of a 3D zonal-type simulation is highly affected by the level of detail of its input data. Static data include the building geometry, construction materials, glazing information, systems used in the building, etc., while dynamic data consist of all time-dependent data such as user-actions (e.g. opening and closing the

* Supported by the BIMERR project funded from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 820621.

windows), occupancy schedules in each of the building zones, use of equipment, weather predictions, etc., commonly being in-building sensed measurements.

In recent studies [4], dynamic data, widely known as Occupant Behaviour (OB), has been criticized as the major cause of uncertainty in building energy performance results. Despite its stochastic nature, in common practice OB is oversimplified and introduced to the 3D zonal-type simulation engines as either deterministic or predefined rule-based schedules, ignoring that way the OB diversity [2]. Hence, having a deeper understanding and properly modeling the occupant behavior have been of paramount importance within IEA EBC Annex 66, where data, methods and models have been developed and applied to understand and reduce the gap between simulated and measured building energy performance by representing OB in a standardized XML schema (obXML).

In the context of the H2020 European project BIMERR,³ the consortium leverages the outcomes of the Annex 66,⁴ adapting the obXML data model to an OWL ontology representation. The new ontology is then used to project occupancy profiles, which mimic building residents actions, and building system utilization boundaries. By means of this representation, occupancy data can be shared among the different applications of the project, fostering semantic interoperability by semantically linking with other domains from the BIMERR ontology network.

In the present work, after explaining the context of the Occupancy Profile ontology within the BIMERR project (Section 2), and the methodology followed (Section 3), the OP ontology, is introduced (Section 4). An example of use is provided (Section 5) before presenting some conclusions and future lines of work (Section 7).

2 BIMERR Project Context

The BIMERR project aims to design and implement a set of Building Information Modelling tools to facilitate the development of building renovations projects. The suite of tools involved in the project includes a decision support system that guides designers in the election of the best renovation scenario taking into account different aspects of the building. These tools range from laser scanners that automatically generate a 3D representation of the existing building to process management tools that are constantly monitoring the planning and execution of the renovation activities. Due to this high diversity of tools, the core of the system need an interoperability layer that support the flow of information between the applications. Such central component data model is driven by a network of ontologies where each modules covers an specific domain within the building sector.

In this context, one of the key modules in the network is the OP ontology, which is utilized to define residents profiles and their interactions with building

³ <https://bimerr.eu/>

⁴ <https://annex66.org/>

system. This information is then needed to make better estimations of building energy performance for each renovation option simulated by the system.

3 Methodological background

The OP ontology has been built following the LOT (Linked Open Terms) methodology, proposed in [8] and further developed at [3]. This methodology focuses on: a) the reuse of already published vocabularies or ontologies and b) the online publication of the built ontology in several formats according to the Linked Data principles.

The LOT methodology defines iterations over a basic workflow composed of the following activities: 1) ontological requirements specification; 2) ontology implementation; 3) ontology publication; and 4) ontology maintenance. The activities, roles involved and expected outputs are depicted in Figure 1.

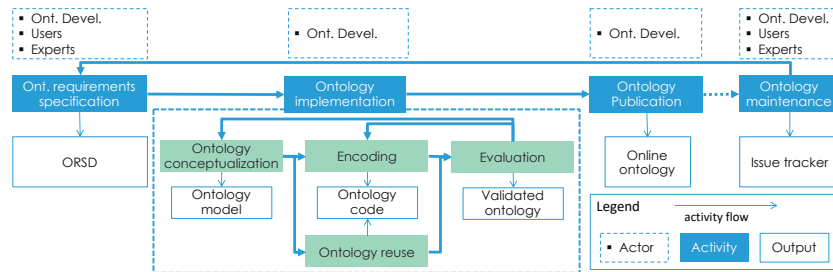


Fig. 1: LOT methodology base workflow. Image adapted from [3].

The goal of the **ontology requirements specification activity** is to collect the requirements that the ontology should fulfill [10]. These requirements are usually grouped as: a) non-functional (purpose, scope, domain, ontology implementation language, etc.) and b) functional (explicit knowledge that should be captured by the ontology). As input for this activity, the LOT methodology proposes domain experts and ontology users (e.g., software developers) to share with the ontology development team documents like manuals, API specifications, data samples, standards, formats used in the community, etc.

Taking as input the ORSD (Ontology Requirement Specification Document), the main goal of the **ontology implementation activity** is to build the ontology using a formal implementation language [10]. LOT proposes to run this phase in different sprints in which a set of requirements are selected to be implemented and an ontology version is produced after each sprint. The ontology implementation is divided into the following sub-activities:

- During the ontology **conceptualization** developers organize and structure the information gathered in the ORSD to build meaningful models at the

knowledge [10]. During this activity ontology developers could use different tools as diagrams (for example, UML based) or description logics to create the model according to the requirements previously acquired.

- The ontology **encoding** activity refers to the translation of the conceptualization into one or more ontologies formalized in OWL.
- Ontology **reuse** refers to the process of using available ontologies for solving different problems [10].
- **Evaluation**. This activity refers to checking the technical quality of an ontology against a frame of reference [10]. For doing so, some evaluation criteria should be selected for example: domain coverage, fit for application, detection of bad practices, logical consistency, etc.) [9] for bad practices detection.

The goal of the **ontology publication activity** is to make the ontology available on-line both as human-readable documentation and in one or more machine-readable formats. Content negotiation mechanisms should be established so that every version of the ontology is reachable from its URI.

Finally, the goal of the **ontology maintenance activity** is to update the ontology after the last release.

4 Occupancy profile ontology development

This section details how each step of the methodological background was applied for the development of the Occupancy Profile Ontology and the particular steps and decisions made to achieve a semantic representation of the domain.

4.1 Requirements Specification for the OP Ontology

The goal of the OP ontology is to provide an ontological representation of the occupant behaviour domain. Therefore, outputs from the Annex 66 project, which aimed to understand the impact of occupant behavior in energy building consumption, were taken as input. More precisely, the following documents were considered during the ontology requirements specification activity: a) the obXML V1.3.3 specification provided by “Occupant Behavior Research at LBNL BTUS”⁵; b) research papers about obXML [6] and [5]; and c) data provided by experts and “Occupant Behavior Research at LBNL BTUS”.

From the obXML and associated documentation, we created three tables with a set of initial functional requirements, following the intermediate representations proposed by METHONTOLOGY. The first table handles the main concepts of the domain, the second table the relationships identified between concepts, and the last table the attributes that describe certain aspect of the concepts by means of literals.

⁵ Available of <https://behavior.lbl.gov/>

4.2 Implementation of the OP Ontology

For this activity, the requirements obtained from the previous step were taken to propose a first draft of the conceptualization. This initial conceptualization, shown in Figure 2, represents concepts as classes, relations between main concepts in the form of object properties, and attributes in the form of datatype properties, providing the model with an OWL representation. Note that for readability purposes, Figure 2 does not show all attributes and terminology described in obXML, but only the main entities. The figure includes also a legend explaining the meaning of the graphics that will be used along this paper.

Even though obXML gives the main resources to model the domain, the constructs available in an XML schema are not directly mapped to OWL constructs, therefore some modelling decisions should be made by ontology engineers to transform obXML into an ontology compliant to the OWL semantics (while keeping a OWL-DL complexity). For this reason, the initial conceptualization in Figure 2 should be reviewed by ontology engineers to make modelling decisions and modifications. Along this section the main Ontological Modelling Decisions (OMD) are described and identified in the following way “(OMDX)” where X is the number of the decision and links the explanation to: (a) the initial situation: numerated dashed (red) ellipses in Figure 2 and (b) current situation: numerated dashed (green) ellipses in Figure 3.

rdfs	http://www.w3.org/2000/01/rdf-schema#
owl	http://www.w3.org/2002/07/owl#
foaf	http://xmlns.com/foaf/0.1/
xsd	http://www.w3.org/2001/XMLSchema#
saref	https://saref.etsi.org/core/
s4bldg	https://w3id.org/def/saref4bldg#
op	https://bimerr.iot.linkeddata.es/def/occupancy-profile#
building	https://bimerr.iot.linkeddata.es/def/building#
time	http://www.w3.org/2006/time#
skos	http://www.w3.org/2004/02/skos/core#

The first decision about ontology engineering was related to the ontology reuse activity. In this case, as no ontology covering the OP domain was known nor found in ontology registries such as LOV,⁶ the decision made was to develop a first ontology conceptualization based on obXML and then to look for subsets of it that could be covered by or aligned to standard or well-known ontologies. In the opposite case, when ontologies covering the domain of discourse are found, the draft conceptualization may be created considering the ontology concepts and relations to be reused. Note that Figure 2 shows some concepts defined in the namespace “building”, which means that those entities come from the Building module of the BIMERR ontology network. For the rest of the paper the namespaces in the box above will be used.

The purpose of the OP ontology is the representation of occupant behavior inside building spaces, and their interactions with energy-related building systems. Hence, the whole conceptualization builds around the root concepts

⁶ Linked Open Vocabularies <https://lov.linkeddata.es>

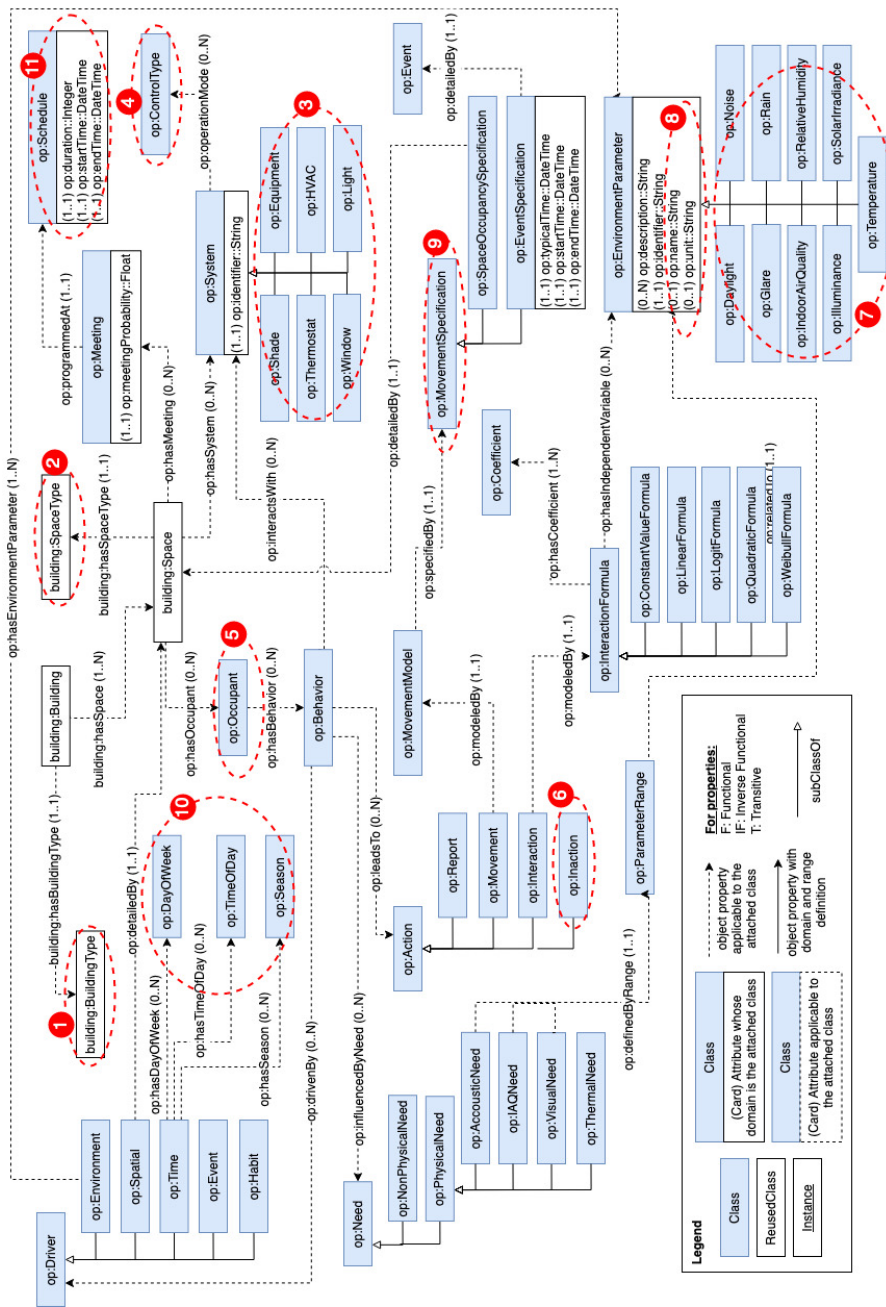


Fig. 2: Initial Conceptualization of the Occupancy Profile Ontology

building, space, behavior, occupant, system, and action. Building and Space are concepts pertaining to the building module of the BIMERR ontology network, however they also play an important role within the occupancy profile context. In the conceptualization we reuse the classes created for the Building ontology,⁷ which at the same time reuse and are aligned with some well know ontologies in the LBD community such as BOT,⁸ or SAREF4BLDG.⁹

The `building:Space` concept was extended in its initial definition to support the specification of the number of occupants inside spaces, feature that is important from the occupancy perspective. This extension involves the attributes `op:maximumOccupants` and `op:minimumOccupants`. It also should be mentioned that we eliminated the initial concepts referring to building types (**OMD1**) and space types (**OMD2**). These concepts were originally considered to support specific taxonomies provided within the obXML schema, however, as RDF(S), and therefore OWL, provides a construct with specific semantics to build taxonomies, we decided to include them as hierarchies of classes under the concepts `building:Building` and `building:Space` respectively (indicated but not added in Figure 3), without losing expressiveness in the model.

A space may contain several systems that occupants can interact with to adjust their comfort requirements. The obXML schema provides an initial list of systems. Instead of creating a new class to allocate such list, we reuse concepts available in SAREF and SAREF4BLDG (**OMD3**). For instance, for HVAC equipment we used the concept `saref:HVAC`, subclass of `saref:Device` and `s4bldg:PhysicalObject`. For building objects not supported in SAREF4BLDG, new concepts are created, for example `op:LightingDevice`. The link between a space and those systems is achieved by the property `s4bldg:contains`, which connects a building space to individuals of the class `s4bldg:PhysicalObject`. The relation between systems and their respective control type is modeled by means of the property `op:hasOperationalMode` (**OMD4**). The obXML documentation provides a list of control options that is represented in the model by the class `op:OperationalModeConcept`, subclass of the concept `skos:Concept`. This modelling decision is recurrent with concepts that gather types. For the `op:ControlType` case, the list of “types” could lead either to a class hierarchy or to an enumeration class. A particular case of the latter is modelling this enumerations in a richer way following the SKOS model [1] as it allows us to relate the created instances by means of the properties `skos:narrower` or `skos:broader` among others. The decision was to keep the main hierarchy based on device functions (e.g. thermostat, light, etc.), as it is a more natural classification than based on the control type, and to model the control types as SKOS as it allows for a richer expressivity than an enumerated class.

obXML describes an occupant as a particular person instead of a general profile. It means, an instance of `op:Occupant` has a particular name and age. For this reason it has been aligned with the `foaf:Person` class. Albeit, attributes like

⁷ <http://bimerr.iot.linkeddata.es/def/building>

⁸ <https://w3id.org/bot>

⁹ <https://w3id.org/def/saref4bldg>

`op:age` are not critical or even used for energy calculation, they would be useful to generate a direct mapping with obXML. As the attribute `op:age` changes constantly over time, we incorporate the additional attribute `op:birthDate` for a more resilient model (**OMD5**). See that this attribute has to be created because the FOAF relation to the birth does not include the year, only day and month, and obXML represents the age, for what the year is needed.

An occupant can have several behaviors with respect to the environmental conditions within a building space. A behavior can lead to an action over a system or a passive acceptance of the uncomfortable conditions. This is modeled by the link `op:leadsTo` that connects to union of individuals of the classes `op:Action` and `op:Inaction` (**OMD6**). Although obXML models `Inaction` as being a children element of `Action`, we preferred to represent them as disjoint classes because the semantic of both concepts is completely different. That is, having “inaction” as subclass of “action” might seem contradictory. The option to link from a behavior to an action or inaction is given by an existential constraint defined for the class `op:Behavior` over the property `op:leadsTo` which can take values of `op:Action` or `op:Inaction`

One important aspect in occupant behavior is the stochasticity involved in human-system interaction. The way to model this interaction occurrence probability is through mathematical formulas, represented in the model by `op:InteractionFormula`. As part of the modelling decisions, we group together mathematical formulas into general templates instead of specifying the exact dimensionality of them. The user of the model can define multidimensional equations through the properties `op:hasCoefficient` and `op:hasIndependentVariable`. The main purpose of the equations is to represent the probability of the occupants to interact with systems given a set of environmental factors. In this context, the range of `op:hasIndependentVariable` is the concept `op:EnvironmentalParameter` that describes certain physical properties directly related to the occupant comfort requirements (**OMD7**). The decision in this case was to reuse the class `saref:Property` that represents qualities of features of interest that can be measured such as temperature, indoor air quality, etc. This decision is aligned with the use of `saref:UnitOfMeasure` (**OMD8**) to indicate the units of `op:EnvironmentalParameter`.

The model should allow to express if the occupant internal needs are influencing in a certain way its behavior. The property `op:influencedByNeed` connects a behavior to individuals of `op:Need`. Needs can be further categorized into: a) `op:NonPhysicalNeed` that gathers all the possible instances that cannot be quantified like privacy or status; and b) `op:PhysicalNeed` that represents comfort requirements that could be expressed by fixed intervals over environmental parameter values. The property `op:definedByRange` connects a need type with the class `op:ParameterRange` to define a customized envelop. The `op:ParameterRange` limits are defined by the attributes `op:minRangeValue` and `op:maxRangeValue`. The range of values are related to a environmental property through the n-ary alike class `op:EnvironmentalParameter`. The physical

needs can be classified into `op:AcousticNeed`, `op:IAQNeed`, `op:VisualNeed`, and `op:ThermalNeed`.

In addition, occupant movements (`op:Movement`) can have an impact over building energy performance. In obXML, the movement models are specified by space occupancy specifications and event specifications, gathered in the initial conceptualization under the class `op:MovementSpecification` (**OMD9**). In the final model, movements are further classified according to whether they represent random movements inside spaces (`op:RandomMovement`), or transitions between spaces (`op:StatusTransition`). The randomness of transition events can be approximated by stochastic models (indicated by the property `op:specifiedBy`) represented by the `op:MovementModel` concept, which gathers probabilistic methods as the `op:MarkovChainModel`. Even though, the obXML implementation uses attributes related to event duration under probabilistic models, we consider them as general information that can be attached directly to a transition event; it means the `op:MovementSpecification` concept is not longer needed in the current model.

A behavior could also be triggered by external factors, like the day of the week, environmental properties or the room itself, which are modelled by the concept `op:Driver`. Although, time can be a driver by itself, it also can be combined with other driver types to indicate when they happen. Taking this double facet into account we need to introduce the class `op:DriverSet`, which allows to combine multiples drivers (`op:Driver`). Furthermore, the concept of time can be expressed in many forms in the context of occupancy behavior (**OMD10**). We could refer to seasons, days of the week or to specific parts of the day. Because we need such diversity in the description of time, we look for terms in the well known Time ontology that could be useful for our representation. From that analysis, we conclude that the concept `time:DayOfWeek` covers the need for pointing to specific days of the week and their combinations (weekends, weekdays, etc), that is, specifying “weekend” would be done by 2 triples, one pointing to Saturday and other to Sunday with the property `op:hasDayOfWeek`. This modelling decision has been taken also for the “all” instances in `op:Season` and `op:TimeOfDay`. The holidays are represented with the attribute `op:onHolidays` to point to the specific days of the calendar. Finally, the modelling of `op:Meeting` (**OMD11**) and their schedule has been simplified by considering meetings themselves as scheduled events that have duration, start and ending time.

As a final step in the implementation, we evaluated the ontology using OOPS!. The set of problems detected were categorized into minor and important issues. The type of errors that fell into the “minor” category were related to missing term descriptions and lack of inverse relationships. According to the OOPS! nomenclature, minor problems are not real problems but can make the ontology more appealing to the final user, therefore, we decide to include descriptions for all the terms. The inverse relations were left for future versions of the ontology, with the idea of not increasing the complexity of the already big model. The set of problems within the “important” section indicated the lack of domain and range definitions in most of the properties. Though this type of

error is not critical for the ontology’s functionality, we decided to include domain and range only on those properties that do not interfere with their definitions.

4.3 Publication and Maintenance of the OP Ontology

For the ontology publication, we followed W3C best practices, setting up content negotiation mechanisms for the OP ontology URI <https://bimerr.iot.linkeddata.es/def/occupancy-profile>. From the URI, the ontology can be retrieved in several RDF serializations, and in a human oriented HTML version generated with Widoco¹⁰, where diagrams, descriptions and an example are included.

In order to keep track of new requirements or updates related to the ontology, we use the GitHub Issues section of the OP ontology repository.¹¹ All the issues requested through the platform will be analyzed by the ontology engineers and domain experts, leading to a new version of the ontology when needed.

5 Occupancy profile ontology in practice

This section shows an example of how to instantiate the OP ontology based on a test case provided with the obXML schema. The prefix “data” is used as the namespace to represent the instances in this example. It is important to mention that the graph representation of the example (Figure 4) has been simplified to improve the readability of the diagram. The complete diagram, the RDF code and the XML example are available at the OP ontology GitHub repository.

In the example, there is a building (`data:Building_1`) that contains an office (`data:S2_Researcher_Office`) shared by two researchers (`data:OC_Researcher_1` and `data:OC_Researcher_2`). The first occupant `data:OC_Researcher_1`, a 21 years old male, has a behavior `data:B_Therm` that predisposes him to set the room thermostat (`data:Thermostat_1`) to a desired value. This action is represented by the instance `data:Interaction_1` of type `op:SetToControlValue`, an its probability of occurrence is described formally by the instance `data:Weibull1DFormula_1`. This formula depends on the environmental parameter `data:EnvironmentParameter_1` that describes the property `op:Temperature`. This behavior is driven by external factors grouped and represented by the instance `data:DriverSet_1` that, in this case, is just conformed by the environment driver `op:Environment_1`.

Apart from punctual interactions with systems, the model is also able to represent occupant movements inside buildings. The second occupant `data:OC_Researcher_2`, has the behavior `data:B_M_Researcher` which leads him to stay in his own office 70% of the time. This occupancy activity is represented by the instance `data:SpaceOccupancy_1`. Furthermore, the occupant also performs several transitions between spaces during his stay at the office. The first

¹⁰ <https://github.com/dgarijo/Widoco/>

¹¹ <https://github.com/oeg-upm/bimerr-occupant-behavior/issues>

transition `data:StatusTransition_1` corresponds to his arrival (`op:Arrival`) at the installations. This transition can be approximated by a custom model represented by `data:CustomProbabilityModel_1` which states the possible time when this transition can happen and the probability of occurrence, which in this case states that at 09:00 am the probability of arrival is 70%. This occupant also interacts with the systems inside his office, where the instantiating process would be the same as before. However, in this case, his behavior is also influenced by thermal comfort needs represented by `data:ThermalNeed_1` where the instance `data:ParameterRange_1` declares what are the maximum and minimum comfort values over the `op:Temperature` property.

6 Related work

One of the existing resources related to occupancy behaviour is the SimModel, a data model expressed in the form of an XSD schema that combines building data (IFC schema) with energy related information (EnergyPlus schema). To accommodate the data needs from EnergyPlus tools, occupant behavioral data is encapsulated within the energy information. A transformation service was built to automatically convert a SimModel XSD schema into their OWL counterpart [7]. Even though the model is represented in OWL, the concepts and properties defined are tool-specific, not being able to be reused by other energy estimation software, lacking the interoperability aimed by this kind of technologies.

Regarding non-ontological resources about occupancy behaviour, one of the most interesting endeavors comes from the Annex 66 project. It aimed at providing a scientific description and clear understanding of energy related occupant behaviours inside buildings, along with a set of research methodologies and simulation tools to bridge the gap between occupant behaviour and the built environment. This work has been materialized in the DNAS (Drivers, Needs, Actions and Systems) framework [6] and an XML implementation of the framework [5]. However, up to the authors' knowledge, no OWL ontology that represents this model is available. For this reason, in this paper the obXML schema has been taken as the main non-ontological input to generate the OP ontology in a format compatible with linked data and semantic web technologies.

7 Conclusions and future steps

This paper has presented the process of developing a semantic model, implemented as an OWL ontology, based on the obXML standard. The ontology is available online together with human-readable documentation and an example of use. Along the process it has been observed that different modelling decisions sometimes are taken for similar problems, for example, whether to reduce the number of triples when generating data or not, or whether to create a class hierarchy or a SKOS schema for modelling types. This work presents a catalogue of 10 well-argued decisions that could serve for other ontology developments.

Main future lines of work are to extend the ontology evaluation techniques used, for example including test verification and the elaboration of additional examples. Authors will also explore the construction of mapping rules in order to transform obXML data into RDF statements automatically. Finally, the connection of the OP ontology with other domains from the BIMERR network needs to be evaluated in order to address overlaps, and to determine how this complementary data (sensor data, building data, etc) can enrich the description and help in the understanding of occupancy profile information.

References

1. Bechhofer, S., Miles, A.: SKOS simple knowledge organization system reference. W3C recommendation, W3C (2009)
2. Cowie, A., Hong, T., Feng, X., Darakdjian, Q.: Usefulness of the obfmu module examined through a review of occupant modelling functionality in building performance simulation programs. In: IBPSA Building Simulation Conference (2017)
3. García-Castro, R., Fernández-Izquierdo, A., Heinz, C., Kostelnik, P., Poveda-Villalón, M., Serena, F.: D2.2 Detailed Specification of the Semantic Model. Tech. rep., Universidad Politécnica de Madrid (UPM), VICINITY Project. <https://vicinity2020.eu> (2017)
4. Hong, T., Chen, Y., Belafi, Z., D'Oca, S.: Occupant behavior models: A critical review of implementation and representation approaches in building performance simulation programs. In: Building Simulation. vol. 11, pp. 1–14. Springer (2018)
5. Hong, T., D'Oca, S., Taylor-Lange, S.C., Turner, W.J., Chen, Y., Corgnati, S.P.: An ontology to represent energy-related occupant behavior in buildings. Part II: Implementation of the DNAs framework using an XML schema. *Building and Environment* **94**, 196–205 (2015)
6. Hong, T., D'Oca, S., Turner, W.J., Taylor-Lange, S.C.: An ontology to represent energy-related occupant behavior in buildings. Part I: Introduction to the DNAs framework. *Building and Environment* **92**, 764–777 (2015)
7. Pauwels, P., Corry, E., O'Donnell, J.: Representing simmodel in the web ontology language. In: *Computing in Civil and Building Engineering*, pp. 2271–2278 (2014)
8. Poveda-Villalón, M.: A reuse-based lightweight method for developing linked data ontologies and vocabularies. In: *Extended Semantic Web Conference*. pp. 833–837. Springer (2012)
9. Poveda-Villalón, M., Gómez-Pérez, A., Suárez-Figueroa, M.C.: OOPS! (Ontology Pitfall Scanner!): An On-line Tool for Ontology Evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)* **10**(2), 7–34 (2014)
10. Suárez-Figueroa, M.C., Gómez-Pérez, A., Fernandez-Lopez, M.: The NeOn Methodology framework: A scenario-based methodology for ontology development. *Applied ontology* **10**(2), 107–145 (2015)