AOAME4FloWare: Ontology-based feature models for context-aware configurations in IoT applications

Arianna Fedeli[1,*], Martin Peraic[2], Emanuele Laurenzi[2] and Andrea Polini[3]

[1]Gran Sasso Science Institute, L'Aquila, Italy
[2]FHNW - University of Applied Sciences and Arts Northwestern Switzerland
[3]Computer Science Division, University of Camerino, Camerino, Italy

Abstract

Feature models play a relevant role in capturing and consolidating knowledge within an IoT application (e.g., Smart Home). They facilitate the representation of the many possible systems and devices and their relationships that can be deployed to support an IoT application. Once this knowledge is crystallized, the feature model becomes a reusable resource for configuring specific context solutions, defined as scenarios (e.g., Smart Home solution for hospitals). However, deriving an appropriate configuration from the designed feature model requires deep knowledge of the targeted IoT application and its scenario requirements (e.g., different requirements appear in a smart home solution for a public hospital vs. a private home). To tackle this challenge, we propose *AOAME4FloWare*. Our ontology-based metamodeling approach empowers the integration of feature models with IoT-context ontologies, harnessing the latter's power to facilitate the configuration of an IoT application based on specific development requirements. The Design Science Research methodology was followed, where IoT-context requirements were derived from a real-world IoT scenario. The proposed artifact has been evaluated on a smart hospital solution, showing that feature model configuration can be supported by exploiting the underlying domain ontologies.

Keywords

Feature Model, IoT-context ontologies, IoT, Model-Driven Engineering, Ontology-based Metamodelling

1. Introduction

The proliferation of Internet of Things (IoT) solutions has revolutionized various sectors, enabling data-driven innovation and efficiency [1]. This impact spans transportation, agriculture, wellness, military, homes, buildings, and more, with diverse IoT applications emerging in recent years [2]. Each IoT application, such as smart homes, must adapt to specific deployment scenarios like private residences, public spaces, homes for the elderly, child monitoring, and hospitals [3, 4]. Enterprises developing IoT solutions face the challenge of managing numerous technical elements, including IoT devices, communication protocols, data management, and

BIR-WS 2024: BIR 2024 Workshops and Doctoral Consortium, 23rd International Conference on Perspectives in Business Informatics Research (BIR 2024), September 11-13, 2024, Prague, Czech Rep.

*Corresponding author.
arianna.fedeli@gssi.it (A. Fedeli); martin.peraic@students.fhnw.ch (M. Peraic); emanuele.laurenzi@fhnw.ch (E. Laurenzi); andrea.polini@unicam.it (A. Polini)
0000-0003-4376-1697 (A. Fedeli); 0009-0001-3657-6787 (M. Peraic); 0000-0001-9142-7488 (E. Laurenzi); 0000-0002-2840-7561 (A. Polini)

processing [1, 5]. They must also cater to diverse IoT scenarios with distinct requirements, making a one-size-fits-all solution impractical.

The Model-Driven Engineering (MDE) approach is well-suited for managing these facets, facilitating the development of similar yet distinct solutions using models as first-class artifacts [6]. Feature models, in particular, capture a domain's common and variable features, serving as a Platform-Independent Model (PIM) in the MDE approach [7, 8]. In the IoT context, feature models effectively capture the heterogeneity of IoT devices and application functionalities, enabling systematic management of IoT applications [9, 10]. This knowledge can be reused to produce customized configurations for specific IoT scenarios, addressing unique customer needs while maintaining a common architecture and core functionalities [11]. From an MDE perspective, this configuration process enhances the PIM to create a Platform-Specific Model (PSM).

Configuring a feature model for a specific IoT scenario requires a deep understanding and proper handling of unique context factors [12]. Each IoT scenario operates within a distinctive context influenced by requirements such as the physical environment, operational constraints, and regulatory standards, significantly impacting solution design and implementation [13]. IoT application knowledge must be combined with context-specific requirements to develop accurate configurations. Experts manually configure models based on customer requirements and scenario constraints, but this process is expensive and prone to errors and omissions [12]. Therefore, leveraging structured, reusable, and verified knowledge is desirable to make deriving scenario-specific IoT solutions less costly and more effective.

To address this challenge, we leverage the FloWare approach [10], which proposes a feature model structure to represent and collect IoT application knowledge. IoT solution artifacts can be developed from this structure, starting with the manual configuration of feature models. This paper presents *AOAME4FloWare*, an innovative approach that uses ontology-based meta-modelling techniques to enable automatic reasoning during the configuration phase. This facilitates feature models' configuration, deriving accurate context-aware PSMs. Our objective is to expedite the MDE deployment-ready IoT solutions significantly.

The proposed approach was developed following the Design Science Research (DSR) methodology [14], and the paper's structure aligns with the distinct phases of DSR. Section 2 introduces feature models and IoT-context ontologies as the main concepts in this paper. In Section 3, the *design and development* phases are reported, which contain the modeling approach for supporting feature model configuration through IoT-domain ontologies and its technological implementation as a working prototype. In Section 4, we demonstrate the validity of the approach through a smart hospital scenario. The related work is discussed in Section 5 and the conclusion in Section 6.

2. Background

Feature models and IoT. Feature models are hierarchical tree representations widely utilized in product line engineering to describe the configurable features of a software system [8]. Feature models define sets of related products with shared features, allowing for variations in specific characteristics [15]. Parent features represent overarching functionalities, while child features specify finer details. Relationships between features—mandatory, optional, or

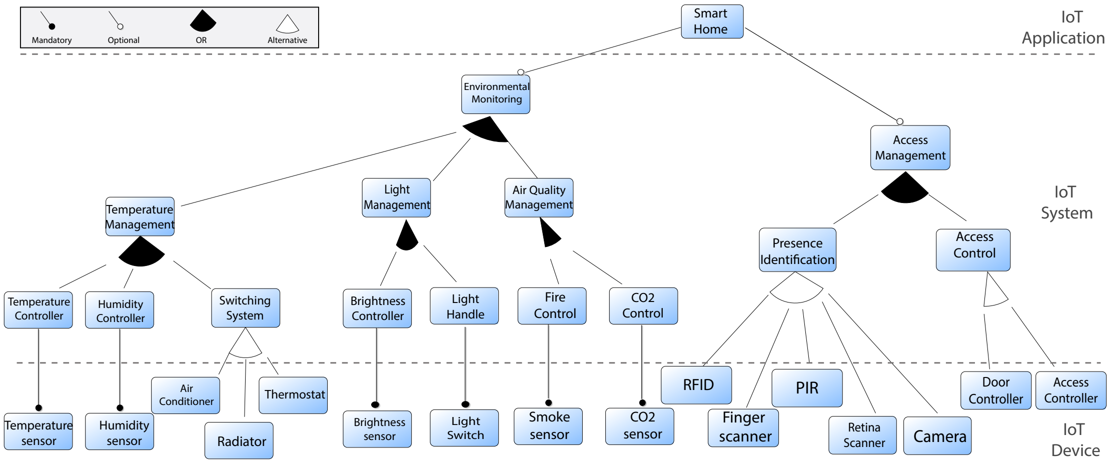

Figure 1: An overview of the Smart Home feature model structure and the legend notation

alternative—are carefully defined by experts to ensure coherence and validity in configurations, reflecting domain-specific knowledge and preventing incompatible feature combinations.

In IoT solution development, feature models streamline the management of features and configurations, allowing developers to handle complexity, variability, and specific requirements across diverse deployment scenarios [16]. These models span various levels of detail, encompassing entire solutions such as Smart Homes [4], Smart Campuses [10], Wireless Sensor Networks, and Body Area Networks [17, 18], as well as individual IoT devices or sets of devices crucial for specific application functionalities [19]. This paper adopts the FloWare approach's feature model structure [10] to capture IoT application knowledge. This MDE strategy supports end-to-end IoT solution development, including feature model design, configuration, and code generation. Figure 1 illustrates a feature model from FloWare tailored for a Smart Home solution. It comprises three layers: the *IoT Application* layer defining the overall context (e.g., Smart Home), the *IoT Systems* layer detailing functionalities (e.g., Environmental Monitoring, Access Management), and the *IoT Device* layer specifying individual IoT devices. The feature model representation offers a robust structure for visualizing and organizing the components of a Smart Home solution, defined as *IoT Application Knowledge*. Additionally, as a feature model property, it ensured that the knowledge acquired could be reused through different configurations made by experts to support various IoT scenarios.

Ontologies and IoT. Gruber [20] defines an ontology as an "explicit specification of a conceptualization," providing a high-level description of concepts and relationships for an agent or community of agents. This emphasizes ontologies' role in fostering a common understanding of a domain. In this work, the domain is the IoT and its applications [21], focusing on specific deployment scenarios. Ontologies, or semantic approaches targeting the IoT domain, conceptualize entities and their relationships, making them machine-interpretable for better query results and automatic reasoning [22]. They are also used to define semantic relationships within IoT architectures and for data integration [23].

Hachem et al. [21] proposed four categories of semantic approaches for IoT ontologies: (1)

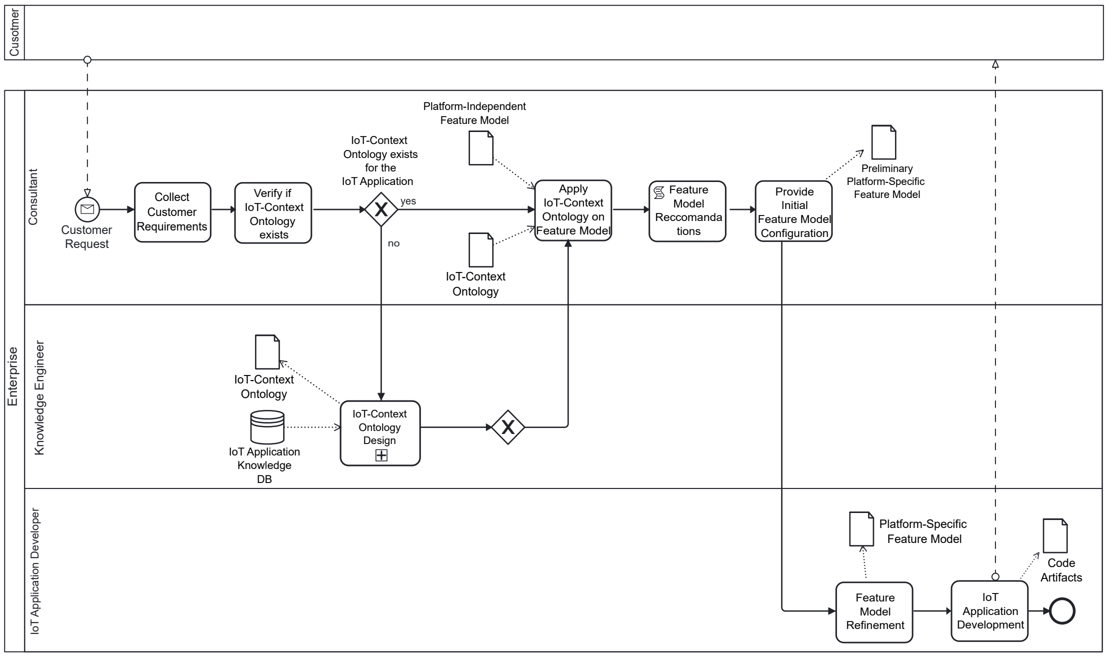

Figure 2: Overview of the proposed AOAME4FloWare approach.

sensors, (2) context-aware, (3) location-based, and (4) time-based ontologies. These categories offer insights into where ontologies are deployed in the IoT domain. Sensor-focused ontologies, like the SSN ontology from the W3C[1], conceptualize aspects of sensor application, interface, and networking, addressing issues like data description, sensor discovery, and data distribution. Domain-specific ontologies focus on particular IoT applications. For instance, the SAREF ontology[2] covers energy management and smart appliances, while the BrickSchema[3] is widely used in smart home and building management, enabling a common understanding for planning and managing smart architectures. BrickSchema uniquely allows the definition of collections, such as systems, to bundle IoT sensors and devices. This paper uses the term *Context-Ontology*, as introduced in [24], to refer to IoT and domain ontologies within the IoT domain.

3. The AOAME4FloWare approach

The proposed approach builds on the FloWare method [10], which models IoT application knowledge using a predefined feature model structure. The FloWare method was developed inside the ADOxx metamodelling platform[4], a development and configuration platform suitable for the development of Domain-Specific Languages. Experts configure a feature model to derive artifacts for desired IoT solutions automatically. However, as discussed in Section 1, the configuration process must manage extensive context information, making it prone to

[1]SSN Ontology: https://www.w3.org/TR/vocab-ssn/
[2]SAREF: https://saref.etsi.org/
[3]BrickSchema: https://brickschema.org
[4]ADOxx metamodelling platform: www.adoxx.org

human error. To address this, we propose *AOAME4FloWare*, an ontology-based solution that (1) grounds feature models in an ontology and (2) maps feature models to domain ontology concepts to support high-quality feature model configuration through automatic reasoning. This approach uses the ontology-based metamodeling method introduced in [25, 26, 27] and extended with reasoning capabilities for ontology-based model validation with SPARQL[5] [28] and ontology-based case-based reasoning [29]. *AOAME4FloWare* supports the end-to-end MDE process for creating deployable IoT solutions, involving multiple steps where different actors produce and refine feature models [30]. Figure 2 illustrates this process, modeled in BPMN 2.0, with the respective actors.

AOAME4FloWare Process Definition. The process begins with a ***Customer*** requesting an IoT solution for a specific application domain from an enterprise ***Consultant***. IoT application domains, like Smart Homes, can encompass diverse scenarios such as hospitals, public spaces, schools, or private homes. The next critical step is gathering requirements from both the customer and the specific application domain. Based on this, the consultant investigates the availability of a suitable *IoT-context ontology* capable of representing the specified scenario. If such an ontology does not exist, the ***Knowledge Engineer*** may need to extend, adapt, or integrate various IoT-context ontologies to meet the requirements fully. If an appropriate ontology already exists, the consultant proceeds to map elements from the ontology-based *Platform-Independent Feature Model* to concepts defined in the IoT-context ontologies. This process results in a model that integrates knowledge from the IoT application (such as systems and devices derived from the feature model) and the specific scenario while remaining independent of implementation platform specifications or technological constraints. Using the ontology-based feature model supplemented with IoT-context ontologies, the consultant can derive from the ontology-based Platform-Independent Feature Model to an ontology-based *Platform-Specific Feature Model.* Next, technology-specific information such as IoT devices and their communication protocols are added to the ontology-based *Platform-Specific Feature Model* by the ***IoT Application Developer***, who is responsible for refining the model into an IoT solution deployable to the IoT platform. The description of both the design of the feature model and the final development of the IoT solutions are neglected in this paper because they are already elaborated in [10].

3.1. Mapping ontology-based feature models with IoT-context ontologies

A mapping between the ontology-based feature models shall be established to take full advantage of the IoT-context ontologies. We rely on the ontology-based metamodeling approach [27] and engineer the metamodel [25, 26]. The benefit of this approach is that the "mappings" are inherited by the model elements because they are class instances of language constructs.

Figure 3 illustrates the ontology-based metamodeling architecture facilitating mapping. The meta-meta layer utilizes RDF(S) as the ontology language for knowledge representation. This language specifies the ontology-based metamodel for the FloWare structure of the feature model. Similarly, IoT-context ontologies are also defined using an ontology language. Mapping occurs in the meta layer between concepts of the FloWare ontology-based metamodel and IoT-context

[5]SPARQL: https://www.w3.org/TR/rdf-sparql-query/

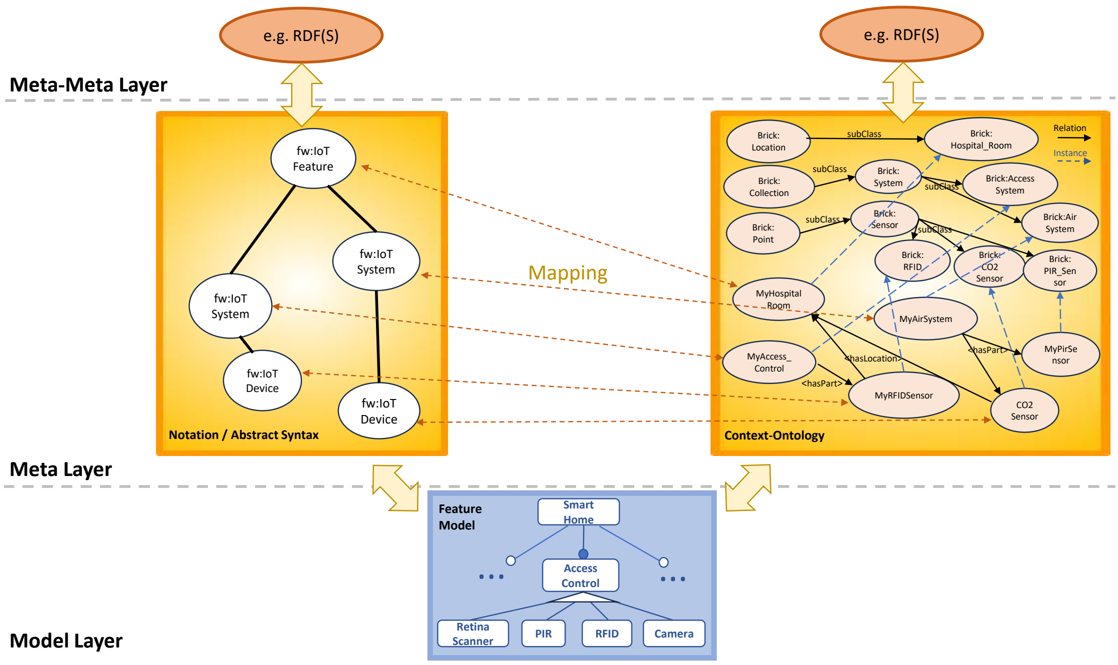

Figure 3: Overview of mapping Feature Model with IoT-context ontologies, adapted from [27]

ontologies. This mapping is inherited in the model layer, where the feature model is depicted at the bottom of Figure 3. Automatic inheritance happens because the model layer contains instances of classes from the meta layer.

To implement the proposed architecture, we extended the AOAME modeling tool [26] to accommodate the FloWare structure of feature models. This resulted in *AOAME4FloWare*.

4. Demonstration of AOAME4FloWare approach

We implemented the approach as a technical prototype using the AOAME tool [26] to demonstrate the approach's feasibility. Focused on a smart hospital scenario within AOAME, we aimed to assess its effectiveness, a critical criterion in evaluating DSR artifacts [31], which measures how well the artifact achieves its intended outcomes. A smart hospital integrates advanced technologies to enhance patient care, operational efficiency, and healthcare services [32]. Initiated by a hospital customer, our scenario aims to enhance patient stay quality through intelligent room solutions. These solutions manage environmental parameters like temperature to effectively minimize virus spread. Leveraging sensor technologies and automation, the smart hospital endeavors to create a safe and hygienic environment for patients, staff, and visitors.

In our approach (Section 3), the scenario begins with a customer's request to develop a smart hospital solution. The consultant evaluates existing IoT-context ontologies for suitability. Since no single ontology fully meets the requirements, the knowledge engineer explores ontologies that can accommodate the necessary representations. Following insights from [33], ontology selection aims to align with the smart hospital's scope, covering entities such as locations, IoT

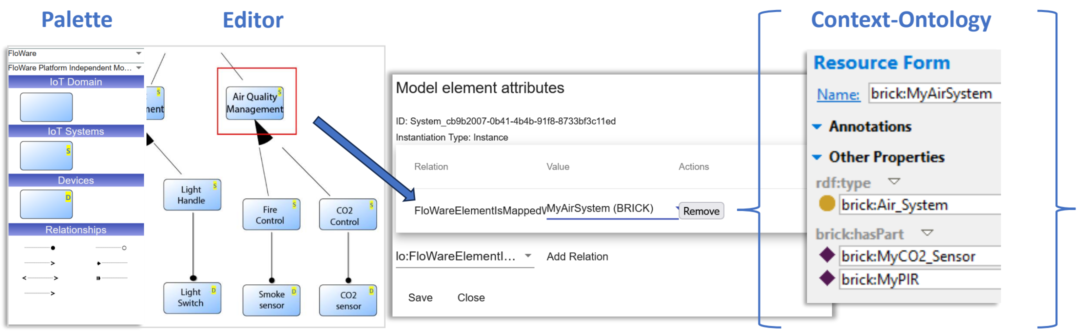

Figure 4: Approach of the Mapping inside AOAME between Notation and Feature Model design (inside the Palette and Editor), Abstract Syntax, and Context-Ontology

systems, devices, and sensors. This approach aims to streamline design efforts by leveraging pre-defined entities and logical groupings from existing IoT-context ontologies. Examples of such ontologies include SAREF4BLDG[6], Project Haystack[7], and BrickSchema[8]. These ontologies offer foundational support for defining and organizing entities pertinent to IoT applications in building management and related domains, closely aligning with our scenario's needs. The *Knowledge Engineer* selects BrickSchema as the ideal ontology for the smart hospital application due to its comprehensive coverage of essential entities like Brick:Sensor and Brick:Equipment, along with robust concepts for organizing them into logical groupings such as Brick:System. However, BrickSchema has not fully met some requirements. Therefore, the *Knowledge Engineer* extends this ontology to address specific needs identified by the customer and the defined application area.

Therefore, in the smart hospital scenario focused on environmental considerations, including a CO_2 sensor in the air quality system of selected hospital rooms is paramount. Guidelines from CDC/HICPAC emphasize that hospitals face increased risks of airborne disease transmission due to poor air quality, although they do not prescribe specific methods for air quality management. Standards such as ANSI/ASHRAE provide foundational principles for ventilation system design and indoor air quality control. ASHRAE specifically notes that higher concentrations of CO_2 indicate lower outdoor air ventilation rates and may increase the risk of airborne transmission, especially in occupied rooms. A semantic rule is developed using SPARQL within the ontology-based approach to address these requirements. This rule defines necessary classes and instances to ensure the integration of a CO_2 sensor into the Air System, as illustrated in Figure 4. The semantic rule establishes relationships using the "Brick:hasPart" relation, linking the Air System with the required CO_2 sensor. Moreover, it is essential to incorporate capabilities for detecting room occupancy to establish different thresholds for CO_2 levels. This may involve integrating additional sensors into the system or linking with other systems capable of accurately detecting occupancy. The ontology extension process continues until all specified requirements for air quality management within hospital environments are fully addressed and integrated into the smart hospital solution.

[6]SAREF4BLDG:https://saref.etsi.org/saref4bldg/v1.1.2/
[7]Project Haystack: https://www.project-haystack.org/
[8]BrickSchema: https://brickschema.org/

a

```
CONSTRUCT {
  ?system mod:isMissing ?missingDeviceSearch .
}
WHERE {
  ?system a fw:System ;
          lo:FloWareElementIsMappedWithBrickConcept brick:MyAirSystem .
  brick:MyAirSystem brick:hasPart ?missingDeviceSearch .

  FILTER NOT EXISTS {
    ?connection lo:modelingRelationHasSourceModelingElement ?system ;
                lo:modelingRelationHasTargetModelingElement ?device .
    ?device a fw:Device ;
            lo:FloWareElementIsMappedWithBrickConcept ?missingDeviceSearch .
  }
}
QUERY RESULTS
mod:System_e989779d-25bd-44ef-80ac-75a26b2ddbe5
        mod:isMissing  brick:MyPIRSensor .
```

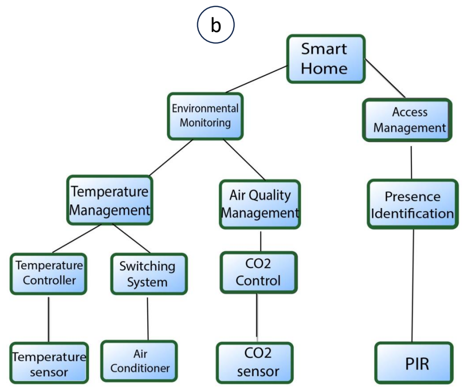

Figure 5: (a) Isolating Unassociated Elements of the MyAirSystem (IoT-Context Ontology to-be) within the FloWare Feature Model (as-is), and (b) the resulting feature model configuration

Once the extended IoT-context ontology is completed, it is uploaded into the AOAME tool for further development and integration into the smart hospital solution. The *Consultant* then engages in ontology-based metamodelling within AOAME, mapping elements from the feature model to relevant IoT context-ontology concepts, as depicted in Figure 4. After the mapping is established, the *Consultant* requests recommendations for configuring the feature model. Leveraging the established mapping, automated recommendations are generated to aid in designing a comprehensive Platform-Specific Model (PSM). For example, in the context of configuring the Air Quality System, the previously developed SPARQL rule (illustrated in Figure 5(a)) is executed. This rule queries instances associated with brick:MyAirSystem from the context ontology, such as instances of brick:CO_2Sensor and brick:PIR as shown in Figure 4. Subsequently, the rule identifies which device instances (fw:Device) within the feature model are linked with systems connected to the brick:MyAirSystem concept in the ontology. This process generates a data graph highlighting any missing components within the Air Quality System, providing crucial context for configuring the feature model. Upon completing the recommendation process, the *Consultant* gains a clear understanding of the finalized IoT solution. For instance, the smart system dynamically adjusts environmental parameters by detecting room occupancy through a PIR sensor. This proactive approach includes managing ventilation rates integrated with the Air Conditioner and monitoring temperature with the Temperature sensor. Furthermore, the solution incorporates a sophisticated CO_2 sensor to assess and maintain optimal air quality within the room continually.

A Preliminary Platform-Specific Feature model is produced at the end of the recommendation activity, as reported in Figure 5(b). To develop IoT solutions, the obtained model must then be enhanced and refined according to specific technological information by the IoT Application Developer. Such a refinement activity is not within the scope of this work, as they have already been discussed and validated in FloWare [10].

5. Related work

Although many research works focus on IoT application development through the MDE approach [34, 35], there is a strong need for reusability mechanisms to enhance the entire development life-cycle [1]. Feature models show positive evidence for reusability and adaptation in expressing IoT domain variability [16, 9]. While some configuration techniques to derive proper feature model configurations have been explored, combining these with ontologies remains underexplored. In [36], a framework is proposed to automatically select suitable features that satisfy stakeholders' functional and non-functional preferences and constraints using group decision-making approaches and Hierarchical Task Network planning. [37] discusses staged configuration for large feature models, where participants iteratively select features, although this can cause conflicts. [38] supports runtime reconfiguration by incorporating context information within each feature, guided by changes in the physical environment and user preferences. Similarly, [39] addresses the challenge of dynamically supporting changes in feature model configurations, using model checking to identify common design faults. [17] proposes a runtime reconfiguration architecture for WSN deployment, using a constraint-checking engine to ensure valid configurations. [40] proposes using artificial intelligence to address functional and non-functional requirements, capturing customer preferences over non-functional properties and considering aspects such as product derivation costs and security.

In the *AOAME4FloWare* approach, we use IoT-context ontologies to guide the experts in this task, reducing stakeholders introducing human errors in the configuration of the feature models, who need a high level of expertise for each IoT scenario while keeping updated on the actual requirements and constraints of that context. The advantage of using these ontologies derives from keeping track of the knowledge of all the requirements and constraints in developing that system in question, allowing, if necessary, to integrate and improve this knowledge with new outgoing constraints easily.

6. Conclusion and future work

The paper highlights the benefit of using feature models to streamline IoT development by systematically representing IoT knowledge for IoT applications (e.g., smart home). However, a massive range of context factors must be considered to configure these models and derive a PSM. We introduced the *AOAME4FloWare*, an ontology-based metamodeling approach to enhance the specific configuration of feature models through context ontologies. By incorporating IoT-context ontologies, *AOAME4FloWare* provides a structured representation of scenario-specific knowledge, encompassing complexities, requirements, regulations, and constraints that must be considered in configuring feature models. As a result, PSMs are expedited, minimizing human errors and benefiting from reusing well-established formal knowledge. We developed a prototype using the Smart Hospital running scenario to validate the feasibility of the *AOAME4FloWare* approach. The results demonstrate the practicality and benefits of using the proposed ontology-based metamodelling approach in developing configured feature models. In future work, we plan to enhance the evaluation of our approach by conducting additional validations with enterprise practitioners to assess its feasibility in real-world contexts. In addition, relevant outcomes from this work suggest AOAME4FloWare could support an ontology-driven configuration of Digital Twins, where IoT elements are defined as their backbone [41, 42]. This could also exploit the possibility of providing a specific modeling language to represent Digital Twins solutions inside modeling environments [43].

References

[1] L. Atzori, A. Iera, G. Morabito, The Internet of Things: A survey, Comput. Networks 54 (2010) 2787–2805.

[2] R. Lohiya, A. Thakkar, Application domains, evaluation data sets, and research challenges of iot: A systematic review, IEEE Internet of Things Journal 8 (2021) 8774–8798.

[3] O. De Ruyck, P. Conradie, L. De Marez, J. Saldien, User needs in smart homes: changing needs according to life cycles and the impact on designing smart home solutions, in: IFIP Conference on Human-Computer Interaction, Springer, 2019, pp. 536–551.

[4] C. Cetina, P. Giner, J. Fons, V. Pelechano, Autonomic computing through reuse of variability models at runtime: The case of smart homes, Computer 42 (2009) 37–43.

[5] M. Singh, G. Baranwal, Quality of service (qos) in internet of things, in: 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU), 2018, pp. 1–6.

[6] D. C. Schmidt, Model-driven engineering, Computer-IEEE Computer Society- 39 (2006) 25–31.

[7] J. Zdravkovic, E. Svee, C. Giannoulis, Capturing consumer preferences as requirements for software product lines, Requir. Eng. 20 (2015) 71–90.

[8] K. Kang, S. Cohen, J. Hess, W. Novak, A. Peterson, Feature-Oriented Domain Analysis (FODA) Feasibility Study, Technical Report CMU/SEI-90-TR-021, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1990.

[9] F. Corradini, A. Fedeli, F. Fornari, A. Polini, B. Re, Floware: An approach for iot support and application development, in: Enterprise, Business-Process and Information Systems Modeling, Springer International Publishing, Cham, 2021, pp. 350–365.

[10] F. Corradini, A. Fedeli, F. Fornari, A. Polini, B. Re, Floware: a model-driven approach fostering reuse and customisation in iot applications modelling and development, Software and Systems Modeling (2022) 131–258.

[11] K. C. Kang, J. Lee, P. Donohoe, Feature-oriented product line engineering, IEEE software 19 (2002) 58–65.

[12] P. Temple, M. Acher, J.-M. Jézéquel, O. Barais, Learning contextual-variability models, IEEE Software 34 (2017) 64–70.

[13] C. Badica, M. Brezovan, A. Bădică, An overview of smart home environments: Architectures, technologies and applications, CEUR Workshop Proceedings 1036 (2013) 78–85.

[14] K. Peffers, T. Tuunanen, M. A. Rothenberger, S. Chatterjee, A design science research methodology for information systems research, J. Manag. Inf. Syst. 24 (2008) 45–77.

[15] A. Q. Gill, V. Behbood, R. Ramadan-Jradi, G. Beydoun, Iot architectural concerns: A systematic review, in: Proceedings of the Second International Conference on Internet of things and Cloud Computing, ICC ’17, Association for Computing Machinery, 2017.

[16] R. T. Geraldi, S. S. Reinehr, A. Malucelli, Software product line applied to the Internet of Things: A systematic literature review, Inf. Softw. Technol. 124 (2020) 106293.
[17] Ó. Ortiz, A. B. García, R. Capilla, J. Bosch, M. Hinchey, Runtime variability for dynamic reconfiguration in wireless sensor network product lines, in: International Software Product Line Conference, volume 2, 2012, pp. 143–150.
[18] A. Venckauskas, V. Stuikys, N. Jusas, R. Burbaite, Model-driven approach for body area network application development, Sensors 16 (2016) 670.
[19] A. Abbas, I. F. Siddiqui, S. U.-J. Lee, A. K. Bashir, Binary pattern for nested cardinality constraints for software product line of IoT-based feature models, IEEE Access 5 (2017) 3971–3980.
[20] T. R. Gruber, Toward principles for the design of ontologies used for knowledge sharing?, International Journal of Human-Computer Studies 43 (1995) 907–928.
[21] S. Hachem, T. Teixeira, V. Issarny, Ontologies for the internet of things, in: Proceedings of the 8th middleware doctoral symposium, 2011, pp. 1–6.
[22] S. N. U. Nambi, C. Sarkar, R. V. Prasad, A. Rahim, A unified semantic knowledge base for iot, 2014 IEEE World Forum on Internet of Things, WF-IoT 2014 (2014) 575–580.
[23] S. Mishra, S. Jain, Ontologies as a semantic model in iot, International Journal of Computers and Applications 42 (2020) 233–243.
[24] C. Emmanouilidis, M. Gregori, A. Al-Shdifat, Context ontology development for connected maintenance services, IFAC-PapersOnLine 53 (2020) 10923–10928. 21st IFAC World Congress.
[25] K. Hinkelmann, E. Laurenzi, A. Martin, B. Thönssen, Ontology-based metamodeling, Studies in Systems, Decision and Control 141 (2018) 177–194.
[26] E. Laurenzi, K. Hinkelmann, A. van der Merwe, An agile and ontology-aided modeling environment, Lecture Notes in Business Information Processing 335 (2018) 221–237.
[27] E. Laurenzi, An agile and ontology-based meta-modelling approach for the design and maintenance of enterprise knowledge graph schemas, Enterprise Modelling and Information Systems Architectures 19 (2024). doi:10.18417/emisa.19.6.
[28] E. Laurenzi, K. Hinkelmann, M. Goel, D. Montecchiari, Agile Visualization in Design Thinking, in: D. R. (Ed.), New Trends in Business Information Systems and Technology, Springer, Cham, 2020.
[29] M. Peter, D. Montecchiari, K. Hinkelmann, S. Gatziu Grivas, Ontology-based visualization for business model design, in: J. Grabis, D. Bork (Eds.), The Practice of Enterprise Modeling, Springer International Publishing, Cham, 2020, pp. 244–258.
[30] P. Patel, D. Cassou, Enabling high-level application development for the Internet of Things, J. Syst. Softw. 103 (2015) 62–84.
[31] N. Prat, I. Comyn-Wattiau, J. Akoka, Artifact Evaluation in Information Systems Design-Science Research - A Holistic View, in: PACIS 2014 Proceedings, 2014.
[32] F. Hasić, B. Beirens, E. Serral, Maturity model for iot adoption in hospitals, Computing and Informatics 41 (2022) 213–232.
[33] J. Park, S. Oh, J. Ahn, Ontology selection ranking model for knowledge reuse, Expert Systems with Applications 38 (2011) 5133–5144.
[34] B. Morin, N. Harrand, F. Fleurey, Model-Based Software Engineering to Tame the IoT Jungle, IEEE Software 34 (2017) 30–36.

[35] F. Corradini, A. Fedeli, F. Fornari, A. Polini, B. Re, L. Ruschioni, X-iot: a model-driven approach to support iot application portability across iot platforms, Computing (2023).
[36] M. Asadi, S. Soltani, D. Gasevic, M. Hatala, E. Bagheri, Toward automated feature model configuration with optimizing non-functional requirements, Information and Software Technology 56 (2014) 1144–1165.
[37] K. Czarnecki, S. Helsen, U. Eisenecker, Staged configuration through specialization and multilevel configuration of feature models, Software process: improvement and practice 10 (2005) 143–169.
[38] J. Mauro, M. Nieke, C. Seidl, I. Chieh Yu, Context-aware reconfiguration in evolving software product lines, Science of Computer Programming 163 (2018) 139–159.
[39] I. de Sousa Santos, M. L. de Jesus Souza, M. L. L. Carvalho, T. Oliveira, E. S. de Almeida, R. Andrade, Dynamically adaptable software is all about modeling contextual variability and avoiding failures, IEEE Software 34 (2017) 72–77.
[40] S. Soltani, M. Asadi, D. Gasevic, M. Hatala, E. Bagheri, Automated planning for feature model configuration based on functional and non-functional requirements, ACM International Conference Proceeding Series 1 (2012).
[41] F. Corradini, A. Fedeli, A. Polini, B. Re, Towards a digital twin modelling notation, in: 2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress, 2022, pp. 1–6.
[42] F. Corradini, A. Fedeli, F. Fornari, A. Polini, B. Re, Dtmn a modelling notation for digital twins, in: T. P. Sales, H. A. Proper, G. Guizzardi, M. Montali, F. M. Maggi, C. M. Fonseca (Eds.), Enterprise Design, Operations, and Computing. EDOC 2022 Workshops, Springer International Publishing, Cham, 2023, pp. 63–78.
[43] D. Karagiannis, R. A. Buchmann, W. Utz, The omilab digital innovation environment: Agile conceptual models to bridge business value with digital and physical twins for product-service systems development, Computers in Industry 138 (2022) 103631. URL: https://www.sciencedirect.com/science/article/pii/S0166361522000264. doi:https://doi.org/10.1016/j.compind.2022.103631.