Building Lifecycle Management System for Enhanced Closed Loop Collaboration

Sylvain Kubler and Andréa Buda and Jérémy Robert and Kary Främling and Yves Le Traon

Abstract In the past few years, the architecture, engineering and construction (AEC) industry has carried out efforts to develop BIM (Building Information Modelling) facilitating tools and standards for enhanced collaborative working and information sharing. Lessons learnt from other industries and tools such as PLM (Product Lifecycle Management) – *established tool in manufacturing to manage the engineering change process* – revealed interesting potential to manage more efficiently the building design and construction processes. Nonetheless, one of the remaining challenges consists in closing the information loop between multiple building lifecycle phases, e.g. by capturing information from middle-of-life processes (i.e., use and maintenance) to re-use it in end-of-life processes (e.g., to guide disposal decision making). Our research addresses this lack of closed-loop system in the AEC industry by proposing an open and interoperable Web-based building lifecycle management system. This paper gives *(i)* an overview of the requirement engineering process that has been set up to integrate efforts, standards and directives of both the AEC and PLM industries, and *(ii)* first proofs-of-concept of our system implemented on two distinct campus.

1 Introduction

Building Information Modeling (BIM) is not a new concept, but rather one that is playing an increasingly larger role in the architecture, engineering and construction (AEC) industry. From design to construction, the concept of BIM has been a feature across many industries for nearly 30 years [19]. It remains a strong and important player in the field because of its ability to allow designers to go beyond representing the physical space of a new or retrofitted building to the intrinsic properties of the structure as well. BIM is not just about the design of new buildings, it also plans for years of use. This is because designing, scheduling, constructing and evaluating a building is done in the BIM model long before any construction actually takes place.

Sylvain Kubler & Jérémy Robert & Yves Le Traon
University of Luxembourg, Interdisciplinary Centre for Security Reliability and Trust, L-2721 Luxembourg, Luxembourg e-mail: `firstname.lastname@uni.lu`

Andréa Buda & Kary Främling
Aalto, School of Science, Department of Computer Science, P.O. Box 15500, FI-00076 Espoo, Finland e-mail: `firstname.lastname@aalto.fi`

Although it is true that the future of the construction industry is digital and that BIM facilitating tools and standards (e.g., IFC) will foster long term facility management, there are still technological and managerial challenges ahead [5]. The nature of these challenges depend on the building lifecycle, which is generally defined as a three-phase process [8]: *(i) Beginning-of-Life (BoL)* including design, manufacture and construction of the building; *(ii) Middle-of-Life (MoL)* including its use and maintenance; and *(iii) End-of-Life (EoL)* including its disposal and recycling. Our research puts special emphasis on post construction challenges.

One of the major challenge after the delivery of the building (i.e., when starting MoL) lies in the difficulty to close the information loop between all phases of the building lifecycle. For example, due to the lack of system integration and other factors such as the non-maturity of the IoT (Internet of Things), it is not that easy to collect, capitalize, and share information/knowledge acquired from MoL (e.g., during use and maintenance activities) with other building lifecycle stakeholders, and *vice-versa* [12]. This is all the more important since such information could result in enhanced decision-making in BoL (e.g., to improve the next generation of buildings and boost the innovation process by capturing new business and user needs), or in EoL (e.g., to guide decision-making about the reuse of components by having information related to the building use conditions). The establishment of such a closed-loop information/collaboration structure throughout the asset lifecycle is not only facing the AEC industry but other sectors, too, e.g., manufacturing where concepts such as *Closed-loop PLM*[1] [15, 14] emerged over the last decade.

Given the above, the contribution of our work is twofold: *(i)* design and develop an open and interoperable building management system that integrates efforts, outcomes (technologies, standards...) and directives of both the AEC industry and adjacent sectors such as Closed-loop PLM and IoT; *(ii)* set up an effective and evolutive requirement engineering framework for ensuring successful system component development. Sections 2 and 3 deal respectively with these two contributions, Section 4 presents proofs-of-concept of our system, the conclusion follows.

2 When BIM meets Closed-loop PLM

The differences between BIM and PLM chiefly surround their capacity for technical and organizational integration. However, they both share a number of similarities relative to their approach to data sharing and project management activities [14]. Although there is only a few documented efforts of implementing PLM in AEC companies, the challenges that follow on from these shared characteristics may provide fertile grounds for sharing lessons learned. Section 2.1 focuses on BIM throughout the building lifecycle, while giving insights into current Closed-loop PLM research and practices. Against this background, section 2.2 discusses the importance of hav-

---

[1] This concept is also referred to as CL_2M: http://cl2m.com

ing accurate requirements to transfer those practices to the AEC/BIM industry, and how this can be achieved using an evolutive requirement engineering framework.

2.1 Whole lifecycle approach

Managing vast amounts of disparate information throughout an asset lifecycle (car, airplane, building...) is an enormous challenge for organizations, particularly in terms of enforcement and compliance. Information governance enforces desirable behavior in the creation, use, archiving, and deletion of corporate information. By closing the loop, rules and policies are defined, policies are managed and enforced, authorized records are accessed when and as needed, and metrics are available to audit the current rules and policies. All this provides a way to continuously assess and update the process for optimum results.

Unlike this vision has been widely explored in PLM, it has only been in the last 5 to 7 years that an increasing focus on the application of BIM throughout the whole building lifecycle has emerged, and the significance of business process integration been acknowledged [14]. BIM servers are now being developed to provide a large integrated data- and knowledge-base that can be leveraged not only in design and engineering, but also in construction operations (BoL), facilities maintenance (MoL), and disposal activities (EoL) [8]. Such a building lifecycle's vision is depicted in Fig. 2, where research efforts are increasingly focused on "closing the loop" to foster collaborative processes, shared resources and decision-making [2]. Although some challenges remains to be addressed in BoL, the major challenge in the context of 'closed-loop' information starts from the delivery of the building, where BIM and other BoL models fall into oblivion. This means that all the knowledge generated in BoL is not, or at least cannot easily be re-used in MoL and EoL, while some reports highlight substantial profits that could accrue from such information loops [13]; for example, Barlish and Sullivan highlight the fact that 85% of the lifecycle cost of a facility occurs after construction is completed [4] and, in this respect, that using BIM-related information in downstream processes could help to save money.

As mentioned above, the concept of closed-loop PLM (or CL_2M) has developed theories and tools to enable closing the information loop between multiple lifecycle phases [15, 11]. This concept emerged from the PROMISE EU FP6 project, where real-life industrial applications required the collection and management of product instance-level information for many domains involving heavy and personal vehicles, household equipment, *etc.* Information such as sensor readings, alarms, assembly, disassembly, shipping event, and other information related to the entire product lifecycle needed to be exchanged between products and systems of different organizations. Based on the needs of those applications, requirements for data exchange were identified and, as no existing standards could be identified that would fulfill those requirements without extensive modification, new messaging interfaces were proposed (see e.g. [11]). Those specifications have since then been further developed by
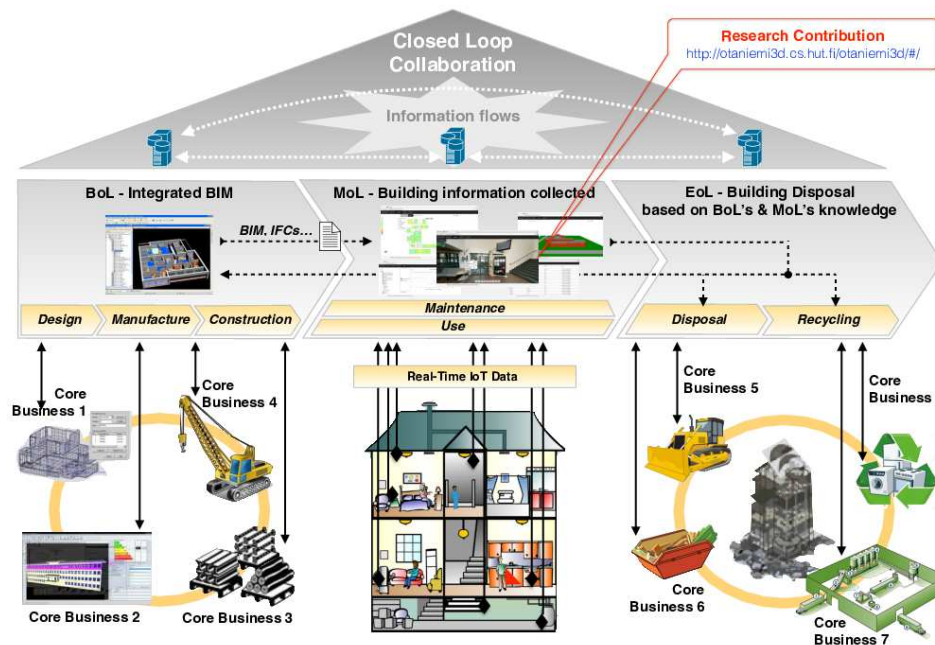
Fig. 1 Building Lifecycle Management System combining BIM and adjacent sectors' efforts

the IoT WG of The Open Group and implemented by several EU project consortia (e.g., LinkedDesign FP7, bIoTope H2020[2]). Recently, The Open Group published those specifications as two distinct – *but complementary* – standards, namely the Open Messaging Interface (O-MI) and Open Data Format (O-DF) standards [11].

Our research work and contribution originate from this state-of-the-art with the specific aim of developing a user-friendly building lifecycle management system relying on open and interoperable standards like O-MI/O-DF. To this end, it is of the utmost importance to select and/or set up an effective and evolutive requirement engineering framework for the development of successful system components, especially in new and cross-domain contexts, as is the case in our study (combination of standards/directives from the AEC/BIM and Closed-loop PLM & IoT sectors). The next section briefly discusses such a requirement engineering framework.

2.2 Requirement engineering framework

Accurate requirements provide the foundation for successful product development. Three main steps can be identified in requirements engineering [16]:

1. *requirements inception:* start the process (business need, market opportunity...);

---

[2] http://biotope-h2020.eu/

2. *requirements development:* include requirements elicitation (consultation with stakeholders), requirements analysis and negotiation;
3. *requirements management*: to capture new needs/contexts over time.

In our context, customer needs must be transferred into product and process requirements without necessarily developing all possible technical characteristics, but only the ones that fulfil the needs for efficient closed-loop information and collaboration in a building's lifecycle context. Let us add that the production activity is supposed to be traceable back at least indirectly to customer requirements.

Given this, our research work develops an hybrid framework based on the synthesis of well established techniques from software engineering and management theory and tools [9]. An overview of this hybrid engineering framework is provided in Fig. 2, which combines (i) a functional analysis (using the Octopus diargam); (ii) a requirement prioritization technique (using AHP – Analytic Hierarchy Process) [18]; (iii) a method that transforms prioritized requirements into quantitative parameters/specifications (using QFD – Quality Function Deployment) [7]; and (iv) a spectral algorithm method for clustering specification conflicts identified through the QFD matrices [3]. These phases are followed up by the software development phase, as well as the definition of KPIs (key performance indicators) to assess whether the system is free of defects, meets the user needs that may evolve over time, and so on. In the context of smart buildings, similar approaches for the development of smart home components was followed, e.g. Durrett et al. [10] used QFD to effectively satisfy customers' needs as part of an integrated smart home environment. Popescu et al. [17] used QFD combined with AHP to identify a set of functions of lighting, heating, security, furniture, *etc.*, that could have a critical contribution to the independent living capacity of people with special needs, if receiving smart abilities. However, none of these frameworks and analyses consider needs related to the whole building lifecycle, along with the imperative to enable closed-loop information and collaboration among distinct lifecycle stakeholders.

As a consequence, an appropriate hybrid engineering framework that enables to integrate such needs, and appropriate technology enablers from the AEC and PLM industries, is defined and proposed in this study, as will be discussed in section 3.

3 A hybrid engineering framework for development of building lifecycle management system components

As previously stated, and depicted in Fig. 2, our framework starts with a functional analysis using the Octopus diagram that identifies the Primary Functions (denoted PF) as well as the Constraint Functions (CF) between the system to be developed and its environment (e.g., actors, directives, services...). Fig. 3 provides insight into the Octopus diagram related to our building lifecycle management system, where the different PF and CF functions are further described in Table 2. Based on these PF and CF functions, high-level requirements are formulated in the form of a hi-
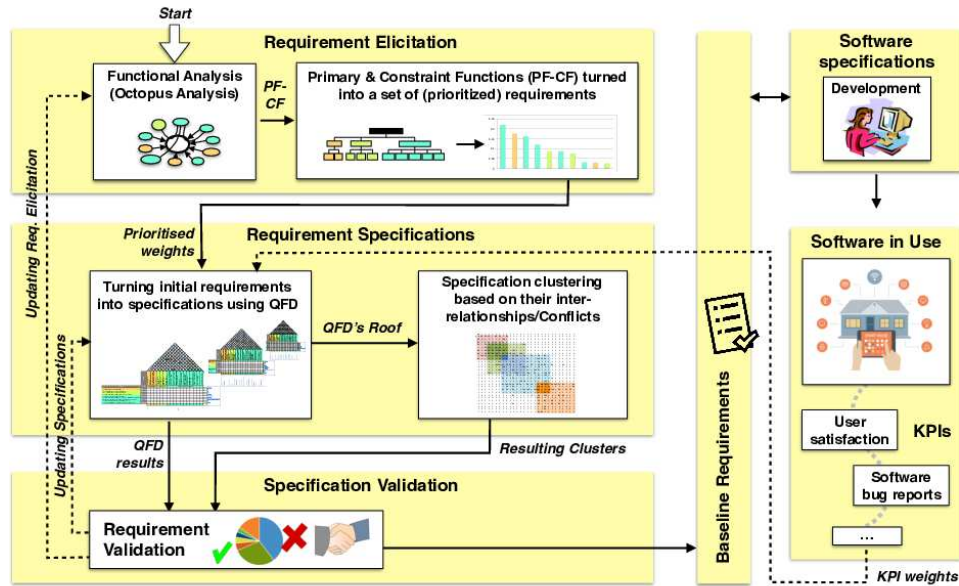
Fig. 2 Evolutive requirement engineering framework for system development

erarchy representing distinct categories (e.g., CF2, CF3 and CF5 falls within the scope of "Building lifecycle"-related requirements...). Due to space limitation, the final requirement hierarchy is not presented in this paper, but a first insight into the categories are illustrated through the Octopus's color code in Fig. 3.

There are number of software requirements prioritization techniques, but according to a recent survey [1], AHP is the most widely used technique. Although we do not present the AHP process in this paper, one should know that AHP provides – *as output* – the list of requirements ranked in order of priority.

Such a requirement ranking, associated with the priority weights, are used as input of the QFD. QFD is both a requirement definition and conceptual design tool that systematically documents customer needs, benchmarks, competitors, and other aspects, and then transforms the list of prioritized requirements into design specifications. The QFD methodology flow involves four basic phases that occur over the course of the product development process. During each phase one or more matrices are generated (*cf.* block in Fig. 2 entitled *"Turning initial requirements into specifications using QFD"*), where the specifications (including their respective weight) resulting from the QFD matrix of phase *n* feed the matrix of phase $n+1$. Fig. 3 provides insight into the specifications resulting from the first QFD matrix, which all bring first technological or scientific enablers to fulfil one or more requirements. For example, Fig. 3 shows that the two most important enablers with respect to our initial requirements are *(i) Data & service discovery:* to enable any building stakeholder to discover and access, when and as needed, information sources and associated knowledge; *(ii) Dynamic service composition:* to enable building end-users to cre-

Table 1 Primary & Constraint Functions (PF-CF) formalized through the Octopus Diagram

| PF/CF	Description
PF1	Enable building stakeholders to easily access and manage various types of data sources (internal or external to the building) according to their role (e.g., maintainers have different information needs than inhabitants).
PF2	Comply with building stakeholders' expectations in terms of Security & Privacy (e.g., some information must be displayed or hidden to a specific category of stakeholders).
PF3	Enable stakeholders to easily create new services based on the integrated data sources and a portfolio of "processing blocks" (including diagnosis, maintenance prediction, event-detection, storage…)
CF1	Ensure the system's scalability (e.g., to dependably integrate new data sources and/or creating new services);
CF2	Make it possible the integration of AEC solutions/standards such as BIM (e.g., IFC, Cobie…);
CF3	Comply with directives affordable plus-energy or nearly zero energy buildings (e.g., Directive 2010-31-EU);
CF4	Provide users with open/ubiquitous GUIs that enable to take into account live stakeholder's preferences, both intuitively and explicitly (e.g., inhabitants habits, preferences, maintainer's needs…);
CF5	Enable live (MoL) simulations based on models made available from BoL (e.g., to identify whether the energy or thermal building's behavior has drifted from initial BoL simulation models);
CF6	Facilitate maintenance of the building lifecycle management system/software in a holistic manner (e.g., reporting of sensor failures to the building manager, of software bugs to developers…);
CF7	Enable interoperability and openness among information systems from the whole building lifecycle;
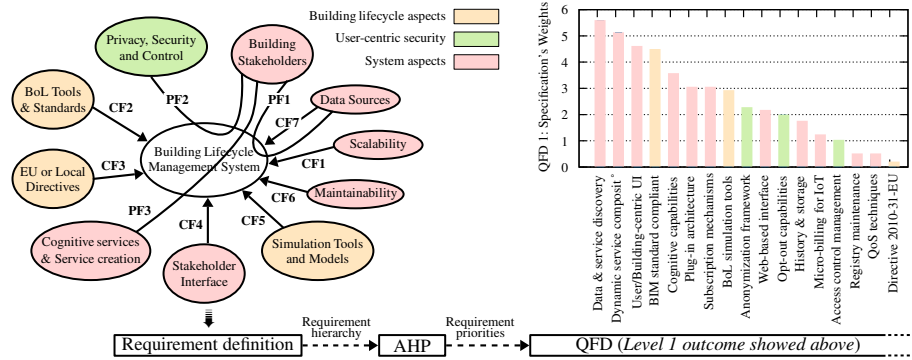
Fig. 3 Illustration of a part of the requirement elicitation & specification steps

ate their own services using a portfolio of "processing blocks" (including diagnosis, maintenance prediction, event-detection, storage…).

As highlighted in our requirement engineering flow (see Fig. 2), a spectral method for clustering conflicts that arise from the QFD matrices' roof[3] is further applied, followed up respectively by a validation phase of the specifications and the development of the software components. To guarantee that our system remains competitive over the short and long term, specific Key Performance Indicators (KPIs) are defined to assess in a quantitative manner several aspects such as the software bugs, the user satisfaction and new needs, *etc*. Although not presented in this paper, it is important to note that these KPI metrics continuously feed the QFD matrices (see Fig. 2), thus helping to produce new releases of the system/software with added or rectified features.

---

[3] The QFD roof emphasis whether two specifications positively or negatively impact on each other.

A building manager is able to specify where a smart object is located in the building/room, and similarly, any end-user can be aware of and access data generated by that object, e.g. he/she can subscribe it, plot it in the Analytics view (see view denoted by ④)
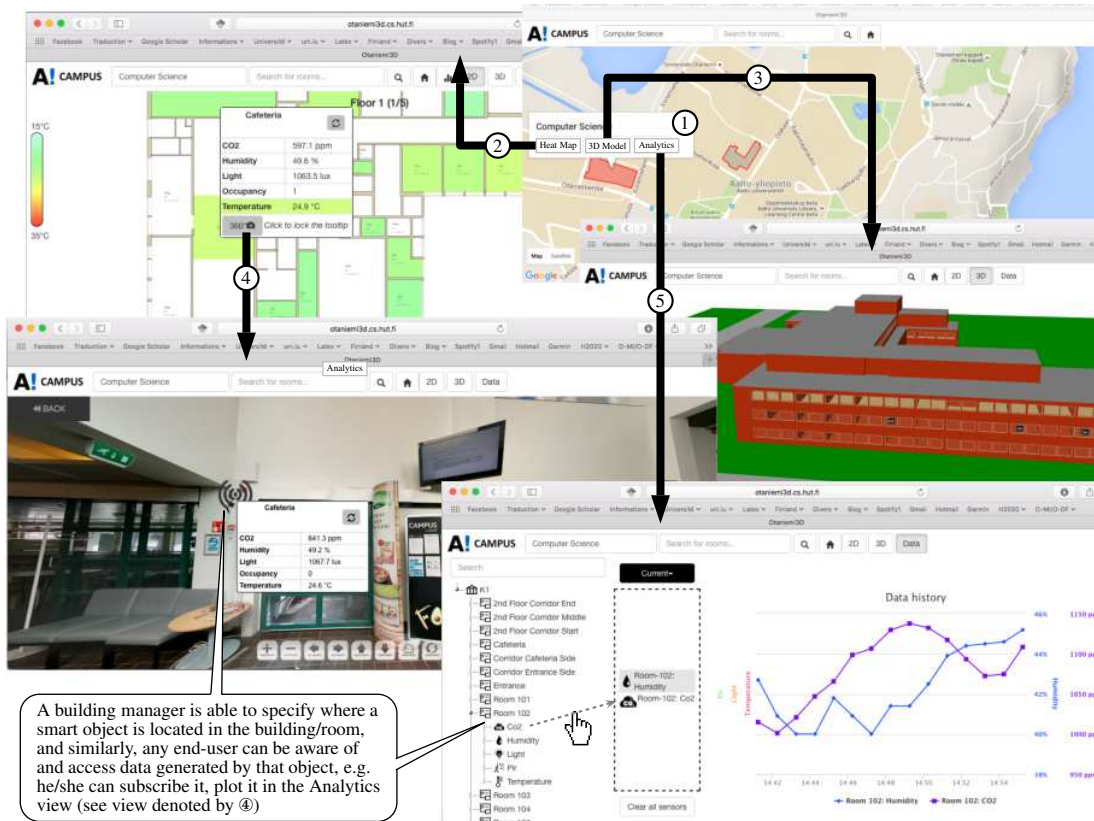
Fig. 4 Instantiation of the Building Lifecycle Management software

4 Proof-of-concept – Building Lifecycle Management System

The first releases of our building lifecycle management software have been supplied[4], enabling any developer to deploy and instantiate it in his/her own environment/buildings. Two proofs of concept of such an instantiation are today available online, namely in Aalto University[5] and Luxembourg University[6].

Fig. 4 provides screenshots of the web-based dashboard that any building end-user/stakeholder can access and use as they see fit. First, stakeholders are able to discover (in a city or region) all the buildings that are compliant with our system, or more specifically compliant with the IoT standards used for data exchange (i.e., O-MI/O-DF in our case). The discovery can be achieved both in a visual manner (Google Map, as shown with the dashboard view denoted by ① in Fig. 4) or auto-

---

[4] https://github.com/AaltoAsia/Otaniemi3D

[5] http://otaniemi3d.cs.hut.fi/otaniemi3d/

[6] http://biotope.sntiotlab.lu:8080/html/snt3d/index.html

mated manner (using the RESTful discovery mechanism supported by O-MI, see e.g. [12]). From this stage, the stakeholder can access both the Floor/2D model (see arrow denoted by ②) as well as the 3D model of the building (see arrow denoted by ③). Those views have been directly generated using – *as input* – the integrated BIM/IFC file, which is now able to be enriched with live sensor data, e.g. the room "Cafeteria" collects Co2, Humidity, Occupancy and Temperature sensor data. Along with this 2D/3D views, another view – *a 360° 's picture of the room (see arrow denoted by ④)* – is supported wherein there is a twofold benefit (i) a building manager can notify the system that a new smart-connected object has been added, but also where it has been added in the room (e.g., a new smart coffee machine), but can also and foremost link the virtual sensor with the real-life information sources (i.e., physical sensors in this example); (ii) any end-user can see where the information source is located, which is a key contextual information that can be further used when developing additional services that rely on or use this information (e.g., if a sensor is located above an oven, the developer can identify it, integrate it to his/her knowledge, and handle it as needed). Finally, our system provides initial Analytics' services (see arrow denoted by ⑤) such as basic plotting of sensor data over a certain period of time regarding one or a group of rooms, but also more complex/cognitive services like the prediction of specific events (energy, failures, *etc.*).

As a final step, we sum up in Table 2 the set of features – *based upon the specifications resulting from the QFD matrix Level 1 (see Fig. 3)* – that have been fulfilled in the first release of our building lifecycle management software (see column denoted "Today's System features" in Table 2), but also the ones that will be addressed through the H2020 bIoTope project and that will be integrated later on (next releases). It can be observed that, in this first software/system release, the specifications having received the highest priorities with respect to the initial requirements (note that the specifications in Table 2 are listed in order of priority).

Table 2 Primary & Constraint Functions (PF-CF) formalized through the Octopus Diagram

QFD Level 1 Specifications	Today's System features	Future features to be developed in the framework of the bIoTope H2020 project
1 Data & service discovery	☁ (supported by O-MI/O-DF standards)	☁ (more advanced/intelligent discovery services)
2 Dynamic service composition	☁ (first widgets – screenshot ① in Fig. 4)	☁ (a more friendly UI & additional widgets)
3 User/Building-centric UI	☁ (see screenshots in Fig. 4)	☁ (context-aware UI: location or role-dependent)
4 BIM standard compliant	☁ (IFC files used as input of our system)	✗
5 Cognitive capabilities	✗	● (Autonomous reasoning capabilities)
6 Plug-in architecture	☁ (requires user interaction)	☁ (new agents specific to building environments)
7 Subscription mechanisms	● (supported by O-MI/O-DF standards)	✗
8 BoL simulation tools	✗	✗
9 Anonymization framework	✗	● (relying on k-anonymity & blurring techniques)
10 Web-based interface	● (see screenshots in Fig. 4)	✗
11 History & storage	☁ (basic database storage)	☁ (new Big Data related capabilities)
12 Micro-billing for IoT	✗	● (digital currency + data quality frameworks)
13 Access control management	✗	● (using XACML or similar standards)
14 Registry maintenance	● (basic log recording)	✗
15 QoS techniques	✗	● (using e.g. Software-Defined Network)
16 Directive 2010-31-EU	✗	✗

LEGEND: ● *Feature fulfilled;* ☁ *Feature partly fulfilled;* ✗ *Feature not fulfilled*

5 Conclusion

It is still challenging to close the information loop between multiple building lifecycle phases (e.g., by capturing information and knowledge from MoL for re-using it in BoL and/or EoL processes), which opens up opportunities for enhanced decision-making and cost saving (e.g., in facilities management). Our research addresses this lack of closed-loop system by developing an open, interoperable and integrated Web-based building lifecycle management system that integrates efforts, directives and technological enablers from both the AEC and PLM/IoT. This paper gives insight into the requirement engineering framework that has been set up for the development of system components, as well as the first proofs-of-concept of the system implementation (running on two distinct campus).

References

1. Achimugu, P., Selamat, A., Ibrahim, R., Mahrin, M.: A systematic literature review of software requirements prioritization research. Information and Software Technology 56(6), 568–585 (2014)
2. Aram, S., Eastman, C.: Integration of PLM solutions and BIM systems for the AEC industry. In: Proceedings of 30th International Symposium of Automation and Robotics in Construction and Mining, Montréal, pp. 1046–1055 (2013)
3. Atkins, J.E., Boman, E.G., Hendrickson, B.: A spectral algorithm for seriation and the consecutive ones problem. Journal on Computing, SIAM 28, 297–310 (1998)
4. Barlish, K., Sullivan, K.: How to measure the benefits of BIM – A case study approach. Automation in construction 24, 149–159 (2012)
5. Azhar, S., Khalfan, M., Maqsood, T.: Building information modeling (BIM): Now and beyond. Construction Economics and Building 12(4), 15–28 (2015)
6. Becerik-Gerber, B., Jazizadeh, F., Li, N., Calis, G.: Application areas and data requirements for bim-enabled facilities management. Journal of construction engineering and management 138(3), 431–442 (2011)
7. Chan, L.K., Wu, M.L.: Quality function deployment: A literature review. European Journal of Operational Research 143(3), 463–497 (2002)
8. Dave, B., Kubler, S., Främling, K., Koskela, L.: Opportunities for enhanced lean construction management using internet of things standards. Automation in Construction 61, 86–97 (2016)
9. De Gea, J.M.C., Nicolás, J., Alemán, J.L.F., Toval, A., Ebert, C., Vizcaíno, A.: Requirements engineering tools: Capabilities, survey and assessment. Information and Software Technology 54(10), 1142–1157 (2012)
10. Durrett, J.R., Burnell, L.J., Priest, J.W.: A hybrid analysis and architectural design method for development of smart home components. IEEE Wireless Communications 9(6), 85–91 (2002)
11. Främling, K., Holmström, J., Loukkola, J., Nyman, J., Kaustell, A.: Sustainable PLM through intelligent products. Engineering Applications of Artificial Intelligence 26(2), 789–799 (2013)
12. Främling, K., Kubler, S., Buda, A.: Universal messaging standards for the IoT from a lifecycle management perspective. IEEE Internet of Things Journal 1(4), 319–327 (2014)
13. Jupp, J.R.: Incomplete BIM implementation: Exploring challenges and the role of product lifecycle management functions. In: 10th IFIP International Conference on Product Lifecycle Management, Nantes, pp. 630–640 (2013)

14. Jupp, J.R., Singh, V.: Similar concepts, distinct solutions, common problems - learning from PLM and BIM deployment. PLM for a Global Market pp. 31–40 (2014)
15. Kiritsis, D.: Closed-loop PLM for intelligent products in the era of the Internet of Things. Computer-Aided Design 43(5), 479–501 (2011)
16. Laplante, P.A.: Requirements engineering for software and systems. In: CRC Press (2013)
17. Popescu, D., Popescu, S., Bacali, L., Dragomir, M.: Home "Smartness"-helping people with special needs live independently. In: International conference of management knowledge and learning and technology innovation and industrial management (2015)
18. Saaty, T.L.: The Analytic Hierarchy Process. New York: McGraw-Hill (1980)
19. Succar, B.: Building information modelling framework: A research and delivery foundation for industry stakeholders. Automation in construction 18(3), 357–375 (2009)