BIM MODEL CONTENT CHECKING: SHORT AND LONG TERM STRATEGIES FOR LARGE BUILDING PORTFOLIO OWNERS

Di Giuda, Giuseppe Martino[a]; Tagliabue, Lavinia Chiara[b]; Locatelli, Mirko[c]; Pattini, Giulia[c]; Paleari Francesco[c]; Campi, Stefano[c] y Tucci, Alessandro[c];

[a]Department of Management, Università degli Studi di Torino, Turin, Italy, giuseppemartino.digiuda@unito.it; [b]Department of Computer Science, Università degli Studi di Torino, Turin, Italy, laviniachiara.tagliabue@unito.it; [c]Department of Architecture built environment and construction engineering, Politecnico di Milano, Milan, Italy, mirko.locatelli@polimi.it, giulia.pattini@polimi.it, francesco.paleari@polimi.it, stefano.campi@polimi.it, alessandro.tucci@polimi.it.

Abstract

Companies define modelling methodology via Building Information Modelling (BIM) protocols. BIM modelling protocols are detailed and documented methods of model production, exchange, and delivery (i.e., a set of modelling step, methodologies and, workflows). Validate and check the compliance between BIM models and modelling protocols is crucial to ensure deliverables are according to expected outcomes. Validation refers to the process of checking models for incompatibility with the defined modelling specifications. Model checking can be a manual or an automated task. The study describes, in the context of a real case study, a model content checking procedure and proposes a comparative analysis between a manual and an automated validation, quantifying the differences in terms of effectiveness and efficiency. Data about time and extent of the two different checking processes are provided. The study aims to identify and measure advantages and limitations of the two procedures in a short- and long-term scenario.

Keywords: *BIM protocols, Regular expression (Regex), Model checking*

Resumen

Las empresas definen la metodología de modelización a través de los protocolos BIM (Building Information Modelling). Los protocolos de modelado BIM son métodos detallados y documentados de producción, intercambio y entrega de modelos (es decir, un conjunto de pasos, metodologías y flujos de trabajo de modelado). Validar y comprobar la conformidad entre los modelos BIM y los protocolos de modelado es crucial para garantizar que los resultados se ajustan a lo previsto. La validación se refiere al proceso de comprobación de la incompatibilidad de los modelos con las especificaciones de modelado definidas. La comprobación de los modelos puede ser una tarea manual o automatizada. El estudio describe, en el contexto de un caso real, un procedimiento de comprobación del contenido del modelo y propone un análisis comparativo entre una validación manual y otra automatizada, cuantificando las diferencias en términos de eficacia y eficiencia. Se aportan datos sobre el tiempo y el alcance de los dos procesos de comprobación diferentes. El estudio pretende identificar y medir las ventajas y limitaciones de los dos procedimientos en un escenario a corto y largo plazo.

Palabras clave: *Protocolos BIM, Expresión Regular (Regex), Comprobación de modelos*

Introduction

Building Information Modelling (BIM) has been widely applied to support the design, engineering, and construction processes (Kensek 2015). Private and public companies, to expand the BIM-data related value to the Operation and Maintenance (O&M) phase, provides modelling protocols to the subject in charge of the modelling task. A critical section of BIM protocols is the definition of modelling methodologies (Cheng 2015), which aim to streamline the data integration of BIM-enabled O&M phase into company's procedures (Edirisinghe 2016). Modeling protocols are detailed and documented methodologies (e.g. steps and modelling methodology workflows) to produce, exchange, and allow access to the consumption of BIM data (Kassem 2014). Moreover, BIM approach cannot be properly exploited into company procedures without checking the consistency and quality of the BIM model against the company's internal modelling protocols.

BIM model checking and validation

The BIM data quality control process relies on a checking and validation procedure i.e., the activity of checking the compliance between BIM models and company's modelling protocols. BIM model checking and validation refer to the process of checking models for incompatibility with the defined modelling specifications ensuring the internal quality and consistency of the model. From this point of view, checking and validation is a process of quality assurance of the model and information content (Hjelseth 2015) defined in the Employer Information Requirement (EIR), and mutually agreed among the parties in the BIM Execution Plan (BEP).

In order to clarify and systematize the BIM checking concept Hjelseth proposed a classification of different types of BIM checking. The classification system separates the compliance checking solutions into:

- Validation checking
- Content checking

Verifying compliance with a predefined rule-set in a BIM-based design is the goal of the validation checking which compares BIM model constraints against pre-defined constraints in the rule-set; clash detection (geometry based) and code compliance checking (information based) are two practical example of validation checking (Hjelseth 2015).

Model content checking aims to check and examine the content of a model for a specified use. Typically, model content checking is a manual activity, performed during the design process. However, model content checking if planned and implemented as a systematic and automatic process can increase the quality of models exploiting the full potentiality of the BIM approach (Hjelseth 2016).

1. Context and scope: RAI information and asset management

The manuscript describes the model content checking procedures applied on the context of a real-case study (i.e., RAI Radiotelevisione Italiana S.p.A. – main Italian TV broadcaster). RAI started implementing the BIM approach in its asset management procedures since 2016 (Di Giuda 2016). During the biennium 2016-2017, a clear and shared set of Asset Information Requirements (AIR) was defined starting from company Organizational Information Requirements (OIR) based on internal asset management procedures. For this purpose, proprietary guidelines and modeling protocols have been defined. For further information on RAI BIM implementation process, the authors suggest to consult the book "Il BIM per la gestione dei patrimoni immobiliari: Linee guida, livelli di dettaglio informativo grafico (LOD) e alfanumerico (LOI)" (Di Giuda 2017).

To reach the goal of optimizing the management of the asset portfolio, BIM models of RAI building asset are being produced. The modelling activity is standardized and coordinated by sharing modelling protocols, procedures, and templates to exchange, and allow access to the consumption of BIM data and streamline BIM-enabled O&M phase into company's procedures, allowing to manage tenders and maintenance contracts. RAI portfolio is highly diversified and consists of several existing assets, planned new construction, and several maintenance and retrofit projects. To accelerate the digitization phase of its portfolio assets, RAI has decided to outsource the modeling activity to external parties.

The outsourcing of the modeling service allows savings in terms of time and costs. However, to ensure that the delivered BIM models are consistent with the information requirements, the delivered models must be checked and validated using predefined procedures.

The study provides a brief description of the checking activity and a comparative analysis of the manual and automated model checking and validation tasks, quantifying the differences in efficacy and efficiency. Short- and long-term advantages and limitations of the two checking and validation procedures are also examined in the study.

2. Methodology

2.1. Content checking: denomination and codification system

As stated, all the BIM models are produced by external design firms in charge of the information modeling of the buildings. BIM models are produced and delivered using the authoring software Revit in .rvt proprietary format to RAI. The denomination and codification system are fundamental for the client to connect BIM data models with the proprietary Database allowing the management of scheduling and recording maintenance operations, storing, and updating technical documents, also giving the possibilities to set up automatic alerts for recurring maintenance and control activities. To achieve a seamless flow of data between BIM models and the management Database the client has required that all objects of the Architectural, Structural, Mechanical and Electrical models must be codified, to univocally identified each object, via the following parameters:

- Settlement_id
- Settlement_code
- Building_id
- Building_code
- Floor_code
- Room_code
- Object_code

All the codification parameters are attributed and must be filled for all objects/instances. In particular, the Object_code must be filled according to the following rules, defined in the modelling protocols:

- System Families (Architectural and Structural models categories: Walls, Architectural and Structural Floors, Roofs, Ceilings, etc.), compiled according to the following rule:
    - <E.T.Code><Number>. With the <E.T.Code> correspondent to the Family Type denomination <E.T.Code>_<thickness cm>_<interior finishing code>_<exterior finishing code>_<Number>.
    - Example: Walls Family type name<CVO_30 cm_RII_RIE_12>, related Wall Type Cod_Object: <CVO12>.

Consequently, RAI requires consistency between specific fields in the object denomination string and the values entered in the codification parameters Cod_Object (e.g., CVO12).

- Loadable families, compiled according to the following rules:
    - Windows instances (Architectural model): progressive code of 2 numeric characters starting from 01 for each floor.
    - Doors instances (Architectural model): progressive code of 2 numerical characters starting from 01 for each room where the door allows access.
    - Instances belonging to other categories (Mechanical and Electrical models): progressive code of 2 numerical characters starting from 01.

Given the importance of compliance with the codification rules, RAI has set the objective of implementing a control and verification system for the content checking of the BIM models produced by external parties. In Table 1 is provided a short description of the related model checking procedures and aim.

Table 1. RAI model checking procedure examined

CHECKING ACTIVITY	DESCRIPTION AND AIM
System families, codification	Verification of the correspondence between the denomination string and the value of the codification parameters <Object_code> of the system families present in the Architectural and Structural model (floors, walls, roofs, ceilings, stairs, and railings).
Loadable families, codification	Verification of the codification parameters value <Object_code> of the loadable families present in the Architectural, Structural, Mechanical and Electrical model.

2.2. **Manual model content checking procedure**

The manual checking procedure of System families consists of comparing the values of the denomination string fields (<E.T.Code><Number>) checking the correspondence with the codification parameters value Cod_Object.

The manual procedure for Loadable families checks the presence of different instances codified with the same codification string (i.e., non-univocal identification of different objects), both the procedures are entire human based.

2.3. **Automated model content checking procedure**

The script to automatically perform the model content checking is developed in Dynamo using Regular Expressions (RegEx). A brief description of Regex origin and development is provided in the following paragraph.

2.3.1 *Regular Expressions: brief introduction and definition*

The term Regex was coined in 1956 by the mathematician Stephen Cole Kleen in his work on finite automata (Kleene 1956) referring to studies in the field of neuroscience conducted by Warren S. McCulloch and Walter Pitts in 1943 (McCulloch 1943). Kleen proposed a mathematical annotation (named set/regular expression) which mathematically represent the McCulloch-Pitts neural model. Kleen studies influenced Ken Thompson who implemented the idea of Regular Expressions inside a text editor called QED (Thompson 1968). QED was the first real case application of RegEx in the field of information technologies and communication.

RegExes belong to the context of formal language theory, a RegEx is a sequence of characters that defines a (possibly infinite) set of text strings (Hopcroft 2001). Regexes are a powerful string manipulation tool, and nowadays all modern coding languages have similar functions. Regular Expressions are employed in data validation, classification, and extraction (Arslan 2005) to verify the structure of a string or process text information performing functions like:

- Searching a string
- Extracting substrings
- Replacing, rearranging parts of a string
- Breaking strings into smaller pieces

At its core, a RegEx is a search pattern composed by a set of symbols that helps to match, locate, and manage text and string data in general. As stated, RegExes are supported by modern programming languages like Python (e.g., Python RegEx module), JavaScript (e.g., JavaScript JS RegExp module), and

C# (e.g., Regex Class). An example of the splitting string function using the RegEx Python module (re package) is provided in Figure 1.

```
#import of the RegEx Python re package
import re

#definition of the string to split
str = 'PVI_85cm_FSR_RII_59'

definition of the function to split string into substrings using the delimitator '_'
substrings = re.split('[_,][_,]',str)

#results
print(substrings)

output: ['PVI', '85cm', 'FSR', 'RII', '59']
```

Fig. 1 String splitting by delimitator, Python code example

2.3.2 *Dynamo RegEx node: output and reporting*

Dynamo nodes have been developed based on RegEx functions, available in the Clockwork package, to automatically check the accuracy of the codification parameters of the System and Loadable families. The Dynamo Clockwork package was chosen because it is one of the few packages that allows to easily implement RegEx functions in Dynamo environment, it is a stable package, constantly updated and based on Python language, the top programming language for PYPL (PopularitY Programming Language index).

To streamline the checking activity, the outputs are exported in a .txt format and imported in a predefined grid which allows to identify possible errors and facilitates the reporting activity. In fact, all the identified codification errors must be communicated to the external parties in charge of modeling to be rectified. The ID (unique IDentifier) of each instance is provided in the report shared to support the design firms in charge of the model updating.

As stated, for System families, RAI requires consistency between fields in the object denomination string and the values entered in the codification parameters Cod_Object (i.e., Wall type: CVO_90cm_RII_RII_CVO12 and Cod_Object: CVO12). The dynamo node for the system families check the consistency between the <E.T.Code> and <Number> fields of the denomination of the families and the values in the Cod_Object providing an output ready for the reporting.

An example of the outputs for a System family (Wall instances) with inconsistency between the denomination fields and the values of the Cod_Object is provided in Table 2.

Table 2. Example of the output of the Cod_object values checking for the Walls category

ID	FAMILY TYPE	COD_OBJECT	TRUE/FALSE
2009837	CVO_90cm_FSR_TES_14	CVO12	FALSE
2010106	CVO_90cm_FSR_TES_14	CVO14	TRUE
2010322	CVO_90cm_FSR_TES_14	CVO14	TRUE
2010916	CVO_30cm_FSR_50	CVO50	TRUE
2011045	CVO_30cm_FSR_50	CVO50	TRUE
2011446	PVI_85cm_FSR_RII_59	CVO51	FALSE
2011971	PVI_85cm_FSR_RII_59	PVI59	TRUE
2012079	PVI_85cm_FSR_RII_59	PVI59	TRUE
2012932	CVO_90cm_FSR_TES_14	CVO14	TRUE
2013375	CVO_16cm_RII_03	CV03	FALSE
2013660	PVI_20cm_RII_RII_23	PVI02	FALSE
2013788	PVI_20cm_RII_RII_32	PVI03	FALSE
2013983	PVI_15cm_RII_RII_24	PVI33	FALSE
2014067	PVI_15cm_RII_RII_24	PVI106	FALSE

The codification of the Loadable families includes a progressive code of 2 numeric characters starting from 01 for each floor for Windows instances, from 01 for each room for Doors instances, and instances belonging to other Categories (e.g., Fire alarms devices in the Architectural model, Mechanical equipment in the Mechanical model, Lighting Devices in the Electrical model etc.) are coded using a numeric progressive code from 01 for all the instances of the category.

The Dynamo node check if in the BIM model are modelled different instances with the same codification string, like:

- Different instances of doors on the same floor and room, same values for the parameter Cod_Floor and Cod_Room, and with the same Cod_Object
- Different instances of windows on the same floor, same values for the parameter Cod_Floor and with the same Cod_Object
- Instances of other categories (Architectural, Mechanical or Electrical) with the same Cod_Object

The model content checking activity on Loadable families allows to control and avoid the presence of different instances codified with the same codification string, an issue for the univocal identification of the objects modelled. In fact, non-univocal identification of different objects, due to codification errors, may interfere with the BIM management of the O&M phase, which is the reason why the checking operation is considered critical by the company.

An example of the outputs for a Loadable family (Doors instances) is provided in Table 3. The door instances have the same Cod_Floor, Cod_Room and Cod_Object and cannot be univocally identified.

Table 3. Example of the output of the Cod_object values checking for the Doors category

ID	FAMILY	COD_FLOOR	COD_ROOM	COD_OBJECT
2964468	ARC_S_2Ante_ORD_LGN	1	210	1
2982904	ARC_S_1Anta_ORD_LGN	1	210	1
4298609	ARC_S_1Anta_ORD_LGN	1	210	1
3750324	ARC_SFC_1Anta_ORD_VET_01	1	206	2
3750329	ARC_SFC_2Ante_ORD_VET_02	1	206	2
3750331	ARC_SFC_2Ante_ORD_VET_02	1	206	2
3750333	ARC_SFC_2Ante_ORD_VET_02	1	206	2
3750822	ARC_SFC_2Ante_ORD_VET_02	1	206	2
3750824	ARC_SFC_2Ante_ORD_VET_02	1	206	2

2.4. **Monitoring and comparison of the manual and automated model content checking**

Both the checking activities were performed manually, and the timing of the task was tracked by collecting data of different BIM models checking (15 models). The tasks include the checking of all the four disciplines: architectural, structural, mechanical, and electrical using the same procedure. The data about the time required to perform a manual activity is then compared to the time to design and run the script required to automate the same checking activities. Based on the data collected on the initial 15 models, the time required to verify all the models that constitute the RAI building assets was estimated.

To compare the manual and the automated model content checking activities comparison criteria have been defined as shown in Table 4.

Table 4. Criteria used to monitor and compare the manual and automated model content checking activity

CRITERIA	DESCRIPTION	U. M.	MANUAL CHECK	AUTOMATED CHECK
C 1)	Time to perform and update the model content checking procedure	[h]	t(1) Time required to perform the manual checking activity	t(1*) Time required to develop the script using RegExes in Dynamo

			t(2) Time required to modify and update the manual checking procedure	t(2*) Time required to maintain or modify and update the script t(3*) Time required to run the script (i.e., perform the automated checking activity)
C 2)	Data sample checked	[%]	d(1) % of the data model checked	d(1*) % of the data model checked
C 3)	Skills and competencies required to the personnel in charge of the model content check	-	s(1) Knowledge of RAI modelling methodologies and protocols s(2) Knowledge of the manual model content checking procedure	s(1*) Knowledge of RAI modelling methodologies and protocols s(2*) Basic skills in Dynamo (i.e., run a node and manage the outputs) s(3*) Advanced skills in Dynamo and Python (e.g. Dynamo node creation and updating, Python scripting and coding, and RegEx language knowledge)
C 4)	Flexibility: feasibility of the checking system to be adapted to changes in modelling methodologies	-	f(1) Flexibility level	f(*1) Flexibility level

2.5. **Short- and long-term scenarios**

Two scenarios, the first involving outsourcing the development of the Dynamo nodes necessary to automate the model content checking and a second scenario involving acquiring within the RAI organization the skills and knowledge necessary for the development and updating of the Dynamo nodes are compared because of the results obtained.

The scenarios comparison supports the identification of short- and long-term strategies for the implementation of BIM model content checking for large building portfolio owners like RAI.

3. **Results**

The results of the criterion C 1), time to perform and update or modify the model content checking procedure, are here provided:

- Manual checking activity
  - Time required to perform the manual checking activity on a BIM model (considering all the Architectural, Structural, Mechanical, and Electrical disciplines): 6 h< t(1) <11 h
  - Time required to modify and update the manual checking procedure: t(2)=15 h
- Automated checking activity:
  - Time required to develop the script using RegExes in Dynamo: t(1*)=175 h
  - Time required to maintain or modify and update the script: t(2*)=15 h

  - Time required to run the script (i.e., perform the automated checking activity on a BIM model, considering all the Architectural, Structural, Mechanical, and Electrical disciplines): t(3*)=1 h

The results of the criterion C 2), percentage of the Data sample checked, are here provided:

- Manual checking activity
  - % of the data model checked (representative sample): d(1): 70% (company-defined minimum verification threshold). Limited to a representative sample.
- Automated checking activity
  - % of the data model checked (representative sample): d*(1): 100%. Extensive, all the data.

The results of the criterion C 3), skills and competencies required to the personnel in charge of the model content check, are provided in Table 5.

Table 5 C 3) skills and competencies required

C 3)	MANUAL CHECK	AUTOMATED CHECK
s(1) Knowledge of RAI modelling methodologies and protocols	Required	Required
s(2) Knowledge of the manual model content checking procedure	Required	Optional
s(2*) Basic skills in Dynamo (i.e., run a node and manage the outputs)	Not required	Required
s(3*) Advanced skills in Dynamo and Python (e.g. Dynamo node creation and updating Python scripting and coding, RegEx language knowledge)	Not required	Optional

The results of the criterion C 4), feasibility of the checking system to be adapted to changes in modelling methodologies, are here provided:

- Manual checking activity
  - Medium feasibility
- Automated checking activity
  - Medium feasibility

4. Discussion

4.1. Comparison criteria results

The results of the criterion C 1) are plotted in the following chart (Figure 2). The graph shows on the x-axis the number of models checked/to be checked (53 BIM models), on the y-axis the time required for the model content checking activity. The orange line represents the time necessary for the control activity carried out manually, while the blue line represents the time for the automated checking. The automated activity intercepts the y-axis at a value of 175 hours (t(1*)), the time required to develop and test the RegEx-based script.

The graph also shows the break-even point (red circle) between the two strategies manual and automated, i.e., the minimum number of models necessary to repay the 175 hours invested for the development of the node due to the higher efficiency of the checking activity. The break-even point occurs with the 23rd BIM model checked.

The automated verification is also more standardized in terms of the time required for the execution. The time necessary to verify 1 model is equal to an hour against a range that varies between 6 and 11 hours, for the manual activity, in function of the dimension of the model and of the experience and ability of the operator. The reduced variability of the checking times allows a better planning of the activity facilitating the respect of the deadlines thanks to the standardization of the process.

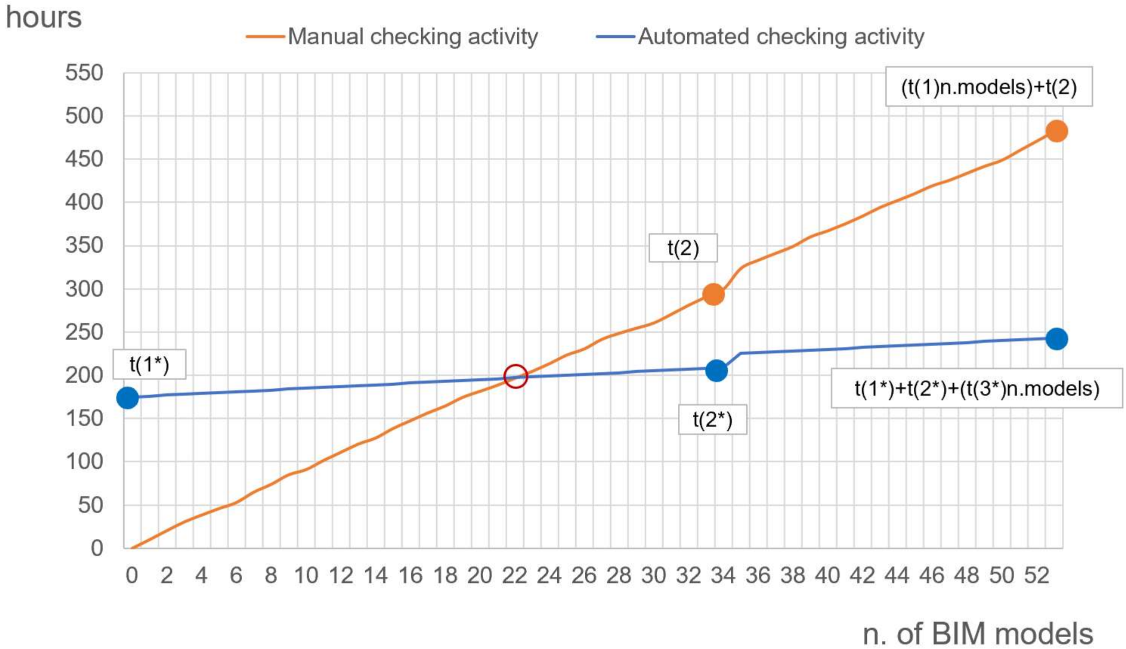

Fig. 2. Manual and automated checking activity comparison chart

Moreover, considering the results of the C 2) criterion, the automated activity allows to check the entire model with a 100% of accuracy, instead using a representative sample for the manual activity equal to the 70% of the data model, defined as the minimum checking threshold by the company. As a result, the manual activity has a lower percentage of data checked and a higher probability of missing errors in the BIM model, on the contrary the automated task has a null percentage of errors.

The results of C 3) criterion highlight the need for more advanced skills and expertise required to the personnel in using or modifying and updating the script used for the automated model checking. In fact, basic skills in Dynamo to run the node and manage the outputs, advanced skills in both Dynamo and Python for the node creation and/or updating, and RegEx language knowledge are required. On the opposite, to perform a manual checking the unique knowledge required to the personnel in charge of the manual activity is about RAI modelling protocols and methodologies, and basic checking procedure.

Both manual and automated procedure have medium flexibility, in terms of feasibility of the checking system to be adapted to changes in modelling methodologies. The time required to modify and update the manual checking procedure and the time required to maintain or modify and update the script are both equal to 15 hours.

4.2. **Scenarios comparison**

As stated, a comparison between the possible outsourcing or acquisition of the competencies and skills to develop the automated checking procedure is proposed in Table 6.

- First scenario, outsourcing the development of the Dynamo nodes necessary to automate the model content checking
- Second scenario, in-house development and updating of the Dynamo nodes necessary to automate the model content checking

Table 6. Outsourcing and axcqusition skill scenarios comparison

CRITERIA	FIRST SCENARIO	SECOND SCENARIO
Personnel training time	Low	High
Availability of the know-how within the company	Low	High
Capacity to maintain and update the script	Null	High
Necessity of personnel training program	Low	High
Necessity of personnel dedicated to the script maintenance and updating	Null	High

Outsourcing the development of the script is considered by the company as a better short-term strategy because external development does not require having highly skilled staff in-house, leading to significant savings in terms of personnel training. In contrast, the in-house script development allows the company to acquire the know-how to develop similar solutions and automate several checking tasks with lower development, maintenance, and upgrade costs in a long-term scenario.

5. Conclusion

The manuscript describes the model content checking procedures applied on the context of a real-case study of a large portfolio owner (i.e., RAI – main Italian TV broadcaster). RAI outsources the modeling activity to accelerate the digitization phase of its portfolio assets with savings in time and costs. To ensure the delivered BIM models are compliant with the information requirements the models have been checked and validate using manual and automated procedures. A brief description of the checking process and a comparison of manual and automated model checking, based on the use of RegEx in a Python/Dynamo environment, are provided in the study. An examination of the short- and long-term advantages and limitations of both checking and validation procedure is also covered in the study.

5.1. **Manual or automated checking?**

The break-even point between the manual and automated strategy (i.e., the minimum number of models necessary to repay the 175 hours invested for the development of the automated checking system) that coincides with the verification of the 23^{rd} checked BIM model highlights the positive return of investment of the implementation of the automatic verification system for the company.

Furthermore, all identified codification errors are communicated to the external parties in charge of modeling to be rectified. To streamline and facilitate the correction activity, a complete report generated semi-automatically is provided to the design firms in charge of the BIM model updating. The report provides the ID (unique IDentifier) of each instance affected by a codification error supporting the external correction activity.

In addition, the error rate for the automated checking is null with a percentage of 100% of the data checked with a lower variability for the time needed to execute the check which allows a better planning of the activity thanks to the standardization of the process.

5.2. **Short- and long-term implementation strategies**

As stated, advanced skills and expertise are required of personnel in using or modifying and updating the script for automatic model checking, which is seen as a limitation by the Client.

In light of the results discussed, the best short-term strategy is to proceed in parallel with the manual model checking while the script/node development is outsourced to a third party. The strategy allows to proceed in parallel with the development of the node without involving internal personnel of the company and with the manual verification of the early models received from the designers.

The acquisition of the competences from the internal staff of the company results to be the best strategy considering the long period. The long-term strategy aims to produce in-house scripts useful for the automation of further verification activities, amortizing time and costs of personnel training making available the technical know-how within the company.

5.3 Next steps

The project wants to continue the implementation of new automated checking procedures and the related monitoring activity investigating possible advantages and disadvantages supporting RAI in short- and long-term strategic choices for the implementation of the BIM methodology in the O&M phase.

Acknowledgements

The authors take the opportunity to acknowledge Umberto Carenzo for the technical support in the development and testing of the Dynamo scripts.

References

ARSLAN, A. N. (2005). "Multiple Sequence Alignment Containing a Sequence of Regular Expressions", in *Computational Intelligence in Bioinformatics and Computational Biology*, pp. 1–7, IEEE. https://doi.org/10.1002/9783527678679.dg08059

CHENG, J. C. P. and LU, Q. (2015). "A review of the efforts and roles of the public sector for BIM adoption worldwide", in *Journal of Information Technology in Construction*, vol. 20, issue July, p. 442–478.

DI GIUDA, G. M., MALTESE, S., RE CECCONI, F. and VILLA, V. (2017). *Il BIM per la gestione dei patrimoni immobiliari: Linee guida, livelli di dettaglio informativo grafico (LOD) e alfanumerico (LOI)*. (Hoepli, Ed.)Milano, Italy

DI GIUDA, G. M., RE CECCONI, F., DEJACO, M. C., MALTESE, S., VILLA, V. and SCHIEVANO, M. (2016). "Information management guideline for asset operational phase", in *Back To 4.0: Rethinking the Digital Construction Industry*, p. 39–49.

EDIRISINGHE, R., KALUTARA, P. and LONDON, K. (2016). "An Investigation of BIM Adoption of Owners and Facility Managers in Australia: Institutional Case study Digital skin of the future construction site View project", in *The RICS Annual Construction and Building Research Conference (COBRA 2015)*, pp. 1–10, Toronto, Canada

HJELSETH, E. (2015). "BIM-based model checking (BMC)", in *Building Information Modeling: Applications and Practices*, pp. 33–61, https://doi.org/10.1061/9780784413982.ch02

HJELSETH, E. (2016). "Classification of BIM-based Model checking concepts", in *Journal of Information Technology in Construction*, vol. 21, issue October, p. 354–370.

HOPCROFT, J. E., MOTWANI, R. and ULLMAN, J. D. (2001). *Introduction to Automata Theory, Languages, and Computation*. (M. Suarez-Rivas & K. Harutunian, Eds.)Second, USA: Pearson Education.

KASSEM, M., IQBAL, N., KELLY, G., LOCKLEY, S. and DAWOOD, N. (2014). "Building information modelling: Protocols for collaborative design processes", in *Journal of Information Technology in Construction*, vol. 19, issue July, p. 126–149.

KENSEK, K. (2015). "BIM guidelines inform facilities management databases: A Case Study over Time", in *Buildings*, vol. 5, issue 3, p. 899–916. https://doi.org/10.3390/buildings5030899

KLEENE, S. C. (1956). "Representation of events in nerve nets and finite automata", in *Automata Studies*, vol. 34, , p. 3–42.

MCCULLOCH, W. S. and PITTS, W. (1943). "A logical calculus of the ideas immanent in nervous activity", in *Bulletin of Mathematical Biophysics*, vol. 5, issue 1, p. 115–133. https://doi.org/10.1007/978-3-030-01370-7_61

THOMPSON, K. (1968). "Programming Techniques: Regular expression search algorithm", in *Communications of the ACM*, vol. 11, issue 6, p. 419–422. https://doi.org/10.1145/363347.363387