The European Master in Building Information Modelling is a joint initiative of:

Univerza *v Ljubljani*

Universidade do Minho

POLITECNICO
MILANO 1863

A Framework For Digital Model Checking

Ahmed Kamal

BIM A+

UMinho | 2020

Universidade do Minho
Escola de Engenharia

Ahmed Kamal

A Framework For Digital Model Checking

BIM A+ European Master in
Building Information Modelling

Co-funded by the
Erasmus+ Programme
of the European Union

September 2020

Universidade do Minho
Escola de Engenharia

Ahmed Kamal

A FRAMEWORK FOR DIGITAL MODEL CHECKING

European Master in
Building Information Modelling

Master Dissertation
European Master in Building Information Modelling

Work conducted under supervision of:
Dr. José Granja
Prof. Dr. Bruno Figueirdo
Ricardo de Matos Camarinha (Tutor in BIMTEC)

Co-funded by the
Erasmus+ Programme
of the European Union

September 2020

AUTHORSHIP RIGHTS AND CONDITIONS OF USE OF THE WORK BY THIRD PARTIES

ACKNOWLEDGEMENTS

I would like to express my special thanks of gratitude to my supervisors Prof. Dr. Bruno Figueirdo and Dr. José Granja as well as Ricardo de Matos Camarinha from BIMTEC who gave me the golden opportunity to conduct this work on Digital Model Checking.

Secondly, I would also like to thank BIM A+ Master Course and Consortium Scholarship for partially covering the tuition fees.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

Ahmed Mohamed

RESUMO

A verificação de modelo digital (DMC) é uma solução que tem o poder de se tornar um ator principal para as preocupações da indústria de AEC. Apesar dos resultados da investigação sobre DMC, ainda existem lacunas para torná-lo prático para resolver problemas do mundo real. DMC, como uma área de investigação emergente, é ainda uma área em desenvolvimento e não completamente formalizada. Isso significa que existe ainda necessidade de aprimorato das capacidades dos sistemas, atualização de processos, ajustes aos atuais documentos de entrega do projeto e padronização adequada dos aspectos de DMC.

O trabalho desta dissertação visa propor uma abordagem de diagnóstico baseada no uso de princípios pré-definidos para analisar o processo de verificação de modelo digital (DMC), um framework formal e um plano de implementação. Esses princípios são o modelo digital de informação (DIM), o conjunto de regras e a plataforma de verificação. Para configurar uma metodologia formal, uma abordagem de modularização foi usada com foco em "o que as coisas são", "qual é a lógica por trás da extensão dos conceitos pré-existentes" e "como isso auxilia o processo DMC". Esses módulos desempenham um papel fundamental e devem ser capturados, verificados e interconectados durante o desenvolvimento da metodologia.

Ao longo da expansão dos princípios, os módulos foram construídos com base em: 1) os DIMs representam a totalidade da informação os quais devem incluir todos sistemas físicos existentes, não apenas os edifícios, 2) as regras de verificação não são apenas originárias de códigos e padrões regulatórios, existindo outras fontes de regras que devem ser levadas em consideração, 3) o papel das partes interessadas envolvidas, sistemas nativos e as fases do projeto não foram ignorados, 4) avaliar a eficácia dos DIMs para integrar, trocar, identificar e verificar seu conteúdo e 5) destacar a existencia de systemas de classificação que poderiam auxiliar no processo de DMC.

Além disso, o DMC é uma atividade dependente que tem causa e efeito nas atividades anteriores e subsequentes. Assim, esta dissertação também propoe um plano de implementação do DMC para se enquadrar nas outras atividades do projeto.

Palavras chave: (Metodologia, Modelo Digital de Informação (DIM) , Processo de verificação, Verificação de Modelo Digital (DMC), Verificação de regras)

ABSTRACT

Digital model checking (DMC) is a solution that has the power to become a primary key player for the AEC industry concerns. Despite the research achievements on DMC, there are still gaps to make it practical to solve real-world problems. DMC, as an emerging research discipline, is still an area of development and not yet completely formalized. This means that there is still a need for enhanced system capabilities, updated processes, and adjustments to the current project delivery documents and proper standardization of DMC aspects.

The work of this dissertation proposes a diagnostic approach based on using pre-defined principles to analyse digital model checking (DMC) and a formal framework and implementation plan. These principles are the Digital Information model (DIM), Rule-set, and checking platform. To set up a formal framework a modularization approach was used focused on “what things are”, “what is the logic behind extending the pre-existing concepts” and “how it assists the DMC process”. These modules play a fundamental role and they must be captured, tracked, and interconnected during the development of the framework.

Throughout the expansion of principles, modules were built on a basis that 1) DIMs are the wholeness of information that should include existing physical systems not only buildings, 2) verification rules are not only sourced from regulatory codes and standards, and there are other sources of rules that should be taken into consideration, 3) the role of involved stakeholders, native system and project phases has not been ignored, 4) evaluate the effectiveness of DIMs to integrate, exchange, identify, and verify its content and 5) highlight on the existent classifications that could aid the DMC process.

Moreover, DMC is a dependent activity that has cause and effect on former and subsequent activities. Thus, this dissertation also proposes a DMC implementation plan that could fit within the other project activities.

Keywords: (Digital Information Models (DIM), Digital Model Checking (DMC), Checking process, Framework, Rule check)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Architecture, Engineering, and Construction (AEC) industries are largely affected by the digital information models (DIM) developed at each stage of a project life-cycle. Users have realized the power of digital models and its authoring environments for applications such as supporting a design, fabrication, shop drawings, quantity take-off and many other purposes serving the industry. Most of these applications require control of information content as an essential phase to make the information management process valid and effective. The industry is still looking for a conventional method of checking information models replacing manual processes, which are costly, laborious and error-prone (W. Solihin and Eastman, 2015). The possibility to manage all the information of a project in an integrated and collaborative manner, during its entire life cycle, can only take place through a key moment, of validation and formal verification, named Digital Model Checking (DMC) that is expected by Young Jr., Jones and Bernstein (2007) to be one of the major contributions to the utilization of information management.

In addition, DMC motivates various stakeholders and involved parties to move towards the higher end of the value chain especially with checks or rules that are hard to interoperate and automate (W. Solihin and Eastman, 2015). In conclusion, Digital model checking (DMC) is a solution that has the power to become a primary key player for the AEC industry concerns. It also allows practitioners and scholars to eliminate time consumption on the engineering debate and unorganized interactions, which often leads to mistrust on the use of technology. Hence, practitioners and scholars are mutually looking forward to using widely a model checking process which provides significant value from both regulatory and industry perspectives. Therefore, this research focuses on the DMC, to maximize the potential values and allow for a wide range of applicability among scholars and practitioners.

Although DMC has potential values, it is still subject to further developments due to relatively numerous constraints. As far as we are aware at the moment of developing this study, there was no proper linking between current determinations that could measure the practicability of current knowledge, and how it could attract non-governmental agencies and public institutes not only scholars and researchers. Moreover, there is a lack of a joint understanding of the different types of checking and the use of terms. This aligns with a wide concentration on “Code Validation and Compliance” seeing it as the wholeness of model checking), although the existence of many other DMC applications, and correspondingly many causes. These deficiencies are correspondingly overextended to standards, classifications and regulatory sources proven by Hjelseth (2016), who searched for related terms resulted in the following list: Compliance checking, Code compliance checking, Clash detection, Rule checking, Model checking, Validation checking, BIM-checking, Quality checking and many more to find that ISO Concept Database (ISO, 2016) does not contain any of the terms above (Hjelseth, 2016).

It is also acknowledged that the support of stakeholders, project phases and available checking platforms is also an inquiry that should be taken into consideration; however, it has not been properly enclosed. DMC has been administered by different checking platforms, and its success largely depends on the willingness to repeatedly invest time and effort in the development of platforms that attract users and increase the platform's base. Nevertheless, the influence of each platform and its ability to fulfil user demands are still indistinct and uncertain. It is also worth mentioning that DMC is not a self-governing process, and in most cases, it is an intermediate station, it is required to ensure the qualitative transformation of information from beforehand activity to an afterward uses.

On the other hand, the assessment of carried-out studied has shown a long historical effort that has been screening the context of Digital Model Checking (DMC) that identifies basic concepts, classifications, and how the rules should be categorized, coded and executed (W. Solihin and Eastman, 2015) (Hjelseth, 2016). Furthermore, the other several trials that have been conducted to translate the textual rules into a machine-readable format while mainly relying on Natural language processing (NLP) similar to the works by Preidel and Borrmann, Zhang and El-Gohary (2012; 2015) and Khan *et al.* (2020). Other researchers have been focusing on DMC for specific domains, such as Health, Safety, and Environment assessment (HSE) (Getuli *et al.*, 2017). This is in-line with several trials of automation of code conformance (Nawari, 2012), automated checking (Lee, 2010), minor efforts on specific project phases (Galkina and Kuzina, 2018).

Despite the persistent research achievements on this research topic, there are still big gaps to make it practical to solve real-world problems. This is to say that DMC, as an emerging research discipline, is still developing and not yet completely formalized. Therefore, this research aims to overview DMC concepts through a classic literature review that will start with a survey of scholarly sources that provides a rough idea on DMC similar but not limited to "Model checking", "BIM-Based Model Checking" "Rule-Based Checking" and "Code Compliance Checking". After that, the methodology of this research will be keen to 1) identify current cornerstones of the DMC process, 2) critically check whether it requires further improvements or not, 3) evolve unimproved concepts, 4) propose strategic approaches, and 5) present patterns to achieve the DMC objective.

1.1. Objectives

The objectives related to this research can be classified into six main parts: the **first** objective is dealing with digital information not only Building information management (BIM). To illustrate, DMC is not only about BIM but also about the wholeness of information that should include existing physical systems not only buildings. The **second is to** justify the current understanding that employs code validation as the only or the most important application of DMC. The **third** objective is to involve the significant role of involved stakeholders along the DMC process while taking into consideration the influence of project phases on DIMs. The **fourth** objective is to evaluate the competence of DIMs to integrate, exchange, identify, and verify its content in a way that supports the best performance of DMC. The **fifth** objective is to overcome the common practice of considering model-checking as an independent activity that has no cause and effect on former and subsequent activities. The **sixth** objective is whether to highlight existing classifications that could aid the DMC process or recommend the presence of additional classification in line with proper documentation.

The expected outcome of these objectives is a modularization approach that will attempt to outline the concepts used along with the development of this research. Also, these objectives are gathered in a DMC implementation plan that outlines the major steps towards an effective DMC process. Furthermore, there are sections following the implementation plan that discuss the associated risks and possible mitigations.

The proposed aims and objective are achieved by a classic literature review, where this review is guided by the following questions:

RQ 1. What is the current body of knowledge about DMCs? How these efforts relate to the AEC industry needs?

RQ 2. What strengthens are required to leverage the current DMC process?

RQ 3. What improvements are needed to the documentation of the DMC process?

The limitation of this research lies in that there were no standards that have been exploring the DMC as a process while considering its milestones, prerequisites and circumstances. This means that there are no guidelines or unified criteria to plan, demonstrate and proper close-out of the DMC process. Also, DMC has never been documented separately and the best the industry has achieved is when DMC has been included within a BIM Execution Plan (BEP) as one of the possible uses of DIMs. This is in line with the absence of any case studies or feedback loop mechanism that could exchange the lesson learned, and try to avoid undertaking the same inaccuracies. Furthermore, the DMC related studies are more theoretical rather than practical that made the goal of the research is quite challenging to be affirmed in the real-world applications to some extent.

1.2. Dissertation Structure

For the sake better-understanding DMC inferences to the AEC industry, this research is being prepared as follows (Figure 1):

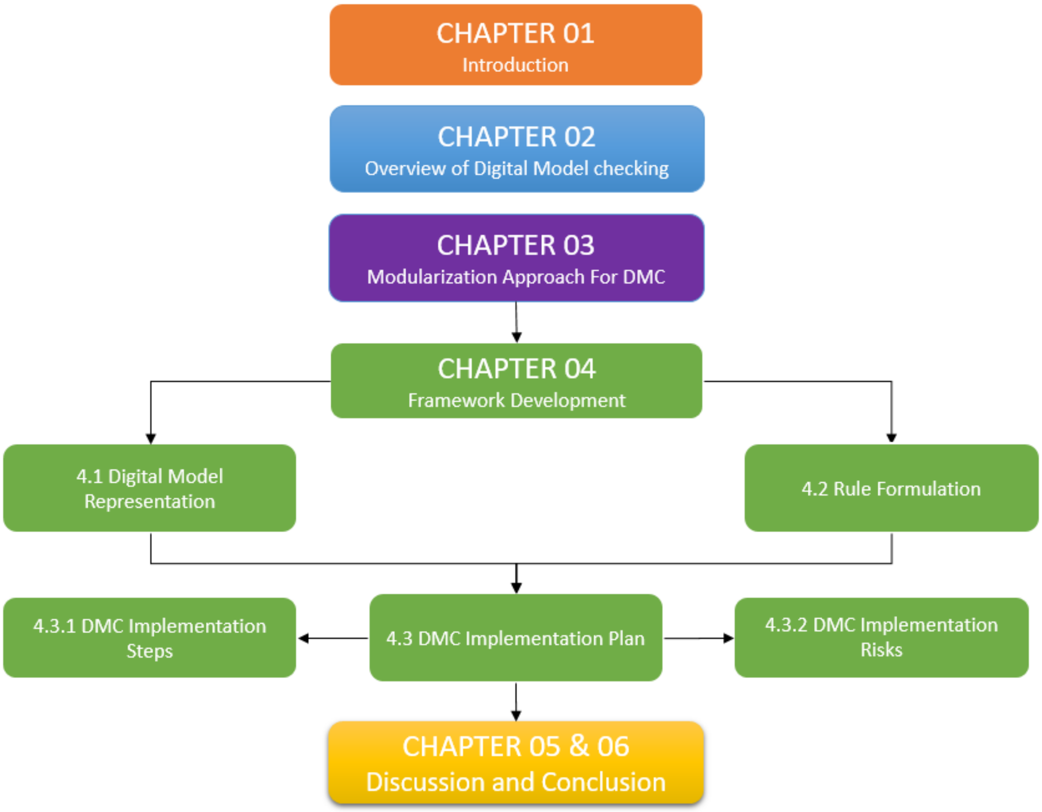

Figure 1 - Structure of the research

- For an introduction (CHAPTER 1), the background of the research is reviewed, the scope and limitations are also acknowledged, and research significance, as well as aims and objectives, are presented.
- The next chapter (CHAPTER 2) overviews DMC by presenting principles of DMC and critically review former efforts supported with a timeline that includes the history of checking platforms and regulations.
- The next chapter (CHAPTER 3) elaborates on how this research has been conducted by presenting designed modules and associated layers to expand the principles of the DMC while clarifying what is the logic behind extending the pre-existing concepts and the possible assistance to the DMC process.
- Afterward, (CHAPTER 4) is detailing the predefined DMC modules and highlight the relationships between the layers in order to accumulate the gained knowledge into a practicable implementation plan that guarantees the success of the proposed framework. The implementation plan will be followed by a list of associated risks.
- Then, this is followed by (CHAPTERS 5 & 6) which include discussion and conclusion.

1.3. Significance of Digital Model Checking

The significance of DMC is an inarguable matter. In line with the raised concerns, this section will present the possible contribution of DMC to the project lifecycle, public services and academy.

Starting with the project lifecycle, McGraw-Hill Construction Research and Analytics (2007) states that higher data sharing is corresponding to higher interest and use in automated checking. The survey has entailed Architects, Engineers, Contractors and owners as main drivers of any project lifecycle, and correspondingly the associated project phase of each driver. Although Architects on average spend almost 50 hours per project on code checking, and 6 % spend more than 100 hours, approximately 85% of architects are positive about working with model checking software to support the code checking process. On the other side, Engineers seem to be leading efforts to enhance the flow of data used for checking. Furthermore, the study claims Engineers have tried DMC nearly twice the rate of architects and owners. Despite the fact that contractors show the least interest, DMC resembles an ideal cost solution for contractors. Eventually, the study claims that authorization is the prime importance to owners that DMC could ensure; however, facility and assist management may also have the same level of importance that DMC can secure.

Most of the organizations and key players within the AEC industry are expecting a process arising from digital model checking, where economic value, quality, and time are core drivers. Value refers to the benefits that an organization can earn by adopting digital model checking, which is normally dependent on capabilities the organization seeks to define and standardize platforms, formats, content, data structure and outcomes serving the needs of owners, authors, and information users down the line. These major outcomes are the core of organizational improvements in profit, business strength, and competitive edge.

For public services institutions, building authorities are pursuing a kind of validation that involves transforming regulations into computable rules for the issuance of building permits (Hjelseth, 2015). Efforts are directed towards automating the process of issuing building permits and assuring a successful national code checking and validation. International Code Council (ICC) in the US took an initiative and tried to fill the gap by generating SMARTCodes in 2006 converting sources using official representations of a few important standards. SMARTCodes provided the jurisdictive form with an authoring tool to manage the alterations of the codes (Dimyadi and Amor, 2013). Unfortunately, SMARTCodes development ended in 2010 due to a lack of funding. The underlying mark-up concept used by SMARTCodes has been further developed by AEC3 (UK) Ltd (Dimyadi and Amor, 2013).

For the academy, scholars introduced DMC researches based on 1) define and breakdown the basic DMC concepts and parameters, 2) review rule checking on specific applications such as health and safety, and 3) translating regulatory and design rules into a machine-readable and executable format through natural language programming via a process named “rule interpretation”.

2. OVERVIEW OF DIGITAL MODEL CHECKING (DMC)

2.1. Principles of Digital Model Checking (DMC)

The scholars who investigated Model-checking has been naming this process as BIM-Based Model Checking; nevertheless, it is a rather narrow approach since checking, as a vital practice, should commonly include the wholeness of information. The Digital Information model (DIM) is probably competent to characterize the existing physical systems as shown in Figure 2 (Cerovsek, 2019). The DIM border is the whole physical building system providing that BIM is existing within the system but not satisfactory demonstrative. To illustrate, this encompasses the digital transformation of any physical building system thru DIMs that involves numerous tools, technologies, and contracts linking the generation and management of digital representations of physical and functional characteristics of information. As a result, this research has adopted a comprehensive notion by using the term Digital Model Checking (DMC) instead of BIM-Based Model Checking.

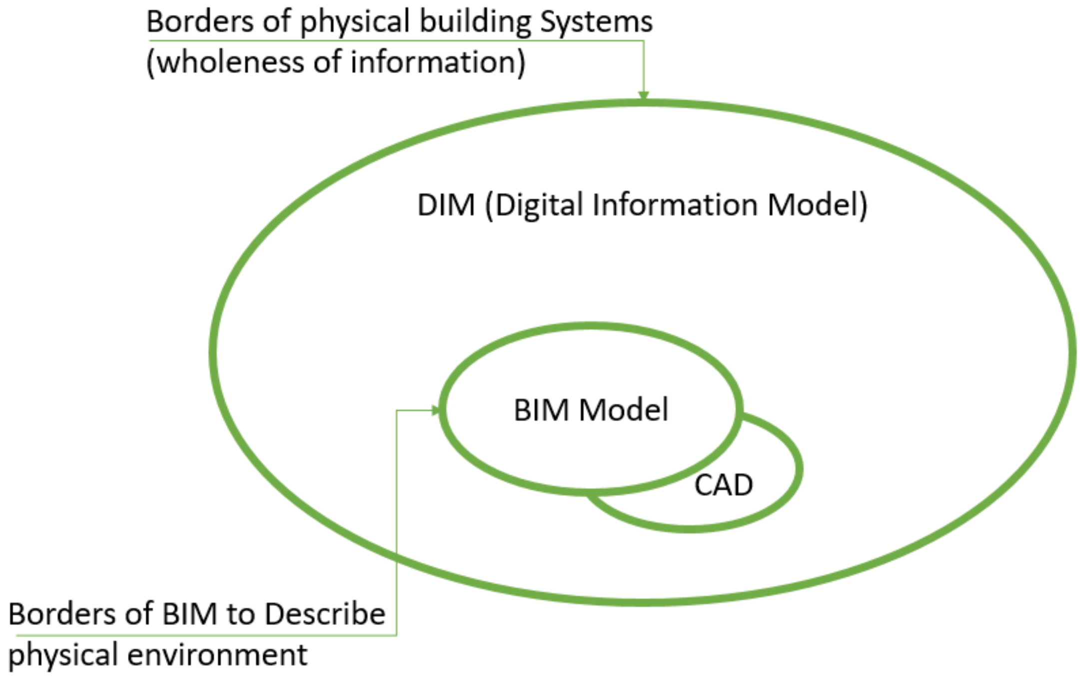

Figure 2 – Digitalization (Source : Cerovsek, 2019)

The current status of Model-checking has proven to be a significant process to verify requirements and design for a variety of schemes to meet a given specification. In this regard, Digital Model Checking (DMC) is prefigured as a process that does not adjust a building design, but rather assesses a design based on relations or attributes providing that given results may be in a form of "pass", "fail" or "warning", or 'unknown' for cases where the needed data is incomplete or missing (Eastman *et al.*, 2009). Despite the given definition, it is worth mentioning that the requirements of DMC may influence the content of DIMs whether by accumulating new information or filtering the existing quantum.

Nowadays, results are not only about reporting since checking platforms have started to the exploit the results of the checking processes by offering several design options in line with codes and regulations which should be manually assessed and verified by the designer or by offering immediate corrective actions similar to update model elements to overcome an exposed inaccuracy.

A correct constructive analysis of DMC principles should be based on the initial explicit conceptual theory. In line with this basic definition, Digital Model Checking (DMC) could be demonstrated as an interaction between three main lanes. These lanes are Rule-set, Digital information model (DIM), and System (e.g. Checking Platform) (Hjelseth, 2015) as shown in Figure 3. Each entity represents a major path that contains some other sub-processes and activities.

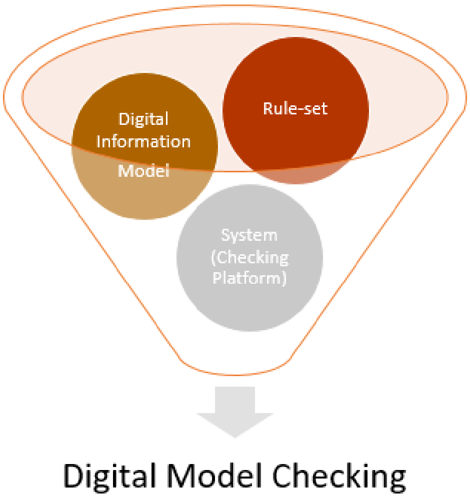

Figure 3 - Principles of Digital Model Checking

-Rule-sets are the instructions generated while analyzing the purpose and use of a DIM to derive assembly of directions controlled by formal rules sources to serves a specific output. For instance, Clash detection, as an example of the possible uses of DIMs, is conducted by given rules that detects a collision of the geometries of two objects or more and controlled by best practices as a control. In general, the majority of rule-sets could be extracted from the following potential examples (W. Solihin and Eastman, 2015):

1. Well-formedness checks. This concerns standards or agreed set of conditions.
2. Regulatory code checking.
3. Best Practices, Exchange Information requirements (EIR), BIM data completeness for handover to the facilities management (FM), and Asset Information Requirements (AIR).
4. Constructability and other logistic requirements.
5. Safety and other rules with possible programmed corrective actions.

-Digital information model (DIM) is the container of both graphical and semantic information where it should be available, accessible, and interoperable.

-System where the textual rule-sets will be converted into a machine-readable language and several checking types could be completed using reliable and efficient platforms and tools.

2.2. A comprehensive summary of previous research on DMC

Digital Model checking in the AEC industry is gaining increased interest due to the wide use of BIM applications throughout project phases. Consequently, digital checking studies have been widely emerging. To enrich this situation, this paper reviews the most recent efforts on DMC studies that focus on the limitations and categorizes the objectives of this researches. The grouping has resulted in four main classes as shown in Table 1.

Table 1 – Summary of recent studies related to DMC

References	Objective	Method Description	Limitations	Class
(Solihin and Eastman, 2015), (Wawan Solihin and Eastman, 2015)	Rule Classification	Classifies rules to their computational complexity and requirements imposed on the rule execution environment.	-Limited to definitions and classifications of rules.	**1**
(Ghannad *et al.*, 2019)	Rule Interpretation	Translates rules from NL-based compliant design requirements to the LRML-based formalized format with VPL-based rule execution.	-A case study that is limited to a specific number of rule conditions (</>/or/and). -No approaches to deal with the complexity of rules. -High contribution required by Software developers.	**2**
(Eastman *et al.*, 2009)	Stages of Rule Implementation	States necessary structure for implementing a functionally complete rule checking and reporting system. Critically review five checking platforms.	-Not clear whether it suits any project phase or not.	**2**
(Zhang *et al.*, 2017)	Rule Interpretation	Designates building codes rules from structured natural language (SNL) to formalized regulations based on SPARQL queries on OWL models.	-A case study that could not be generalized on different classes of rules. -High contribution required by Software developers more than Engineers.	**2**
(Hjelseth, 2016), (Hjelseth and Nisbet, 2010)	Checking Types	Proposes an ontological approach to developing a framework to classify types of model checking and corresponding results.	-The relation or influence on checking platforms has not been revealed. - No explanation on how sources could affect checking types.	**1**

References	Objective	Method Description	Limitations	Class
(Nawari, 2012)	Rule Interpretation	Automates Code Compliance (SmartCodes) Framework using LINQ with XML Programming.	-Limited to cover building code compliance. -High contribution required by Software developers more than engineers. -No methods to deal with further sources such as best practices or EIRs.	**2**
(Fan, Chi and Pan, 2019)	Stages of Rule Implementation	Develops a user-oriented interface to enable users to establish the rules to meet their needs and how the applied rules affect the dependency among BIM model elements.	-A case study that could not be generalized on different classes of rules.	**2**
(Preidel and Borrmann, 2015)	Rule Interpretation	Automates code compliance based on visual programming language	-Limited to cover building code compliance. -High contribution required by Software developers more than engineers. -No methods to deal with further sources such as best practices or EIRs.	**2**
(Khan *et al.*, 2020) **,**(Getuli *et al.*, 2017)	Rule execution on certain Applications	Implement design rules on certain applications (ex. Fire & life safety, energy efficiency, etc.) using language programming	-Limited to specific uses that could not be generalized.	**3**
(Galkina and Kuzina, 2018)	Model-checking and verification at the construction stage	Presents the BIM-models cycle, background, and basic principles of verification system with tools for performing the rule check and the sequence of Construction-BIM model check.	-No framework to determine the specifications of exchanged information and the finest utilization could be obtained.	**4**
(Mekawy and Petzold, 2018)	Model Checking in the early design phases	develop a rule-based expert system to support an automated review of precast concrete requirements in BIM models in the early design stages	No framework to determine the specifications of exchanged information and the finest utilization could be obtained.	**4**

- **Class 1: Fundamental definitions and ontologies of DMC**

Objective:

This class appeals to theoretical definitions and classifications of DMC, rules and checking types supported with some straightforward examples to illustrate the presented modules. (Solihin and Eastman, 2015) presented a significant classification for rule-sets that serves the analytical approach of the rule exploitation and requirements. Hjelseth (2016) explored the types of checking the considering structure of rules, derived logic and potential outcomes. These efforts possibly will be the basis of uniform classification that could be benchmarked and adopted in the future by international organizations for standardization.

Limitations:

Further examinations are mandatory to evaluate these ontologies and therefore decide the feasibility and trustability of forthcoming practices.

Class 2: Stages of Rule Implementation.

Objective:

Almost all efforts in rule checking to date have been applied to building code and accessibility criteria (Eastman *et al.*, 2009). References on **Class 2** generally dispute stages of implementing a rule-set in the DMC process. Eastman *et al* (2009) introduced a formalized rule implementation plan that reflects four main phases commenced by rule interpretation, flowing with the path of preparation and execution phases then ending-up by checking results. Some researches of this class intend to implement a rule in a DMC process and assume completion of the checking process. However, more attention has been given to one of the implementation stages which is the “rule interpretation” whereas the rules are typically written in human-oriented languages that require being transformed into readable machine-language. In this respect, Nawari (2012) has offered automation of code Compliance check thru a framework using LINQ with XML Programming to emphasize the vital statement that once the rule structure has been encoded, it is available for multiple similar uses.

Limitations:

This class seems to be limited to scholars and software developers more than Engineers or Architects as it mandates significant domain knowledge in order to “interpret” rules into a machine-interpretable manner. As a result, expert knowledge is often required to interpret the meaning of the rules; therefore, the majority of AEC drivers are relying on checking platforms more than programming methodologies. One of these commonly and commercially used platforms is Solibri Model Checker (Solibri, 2016), which was intended to be a validation and optimization tool for digital building models, but it, unfortunately, authorizes checks through geometry-oriented rules (Preidel and Borrmann, 2015) or well-defined entities and their relationship that often does not reach the satisfactory level of the users. Moreover, this class has been merely focusing on a limited category of sources such as regulatory codes. In other words, the absence of sources similar to EIR or AIR is often noticeable. Although code compliance is one of the major applications of DMC that researches have been focusing on, there is a misused fact about the nature of regulatory codes as dynamic documents by the means of being updated frequently to deal with changes, potential risks and hazards in the AEC industry, and to compete with the growing technological expansions. Methodologies on Class 2 may have unexploited that rule interpretation is a repetitious process, and these textual updates should be traceable. As long as any standards representation can be linked to the source legal documents, perhaps making use of artificial intelligence (AI) and machine learning techniques may provide a method of keeping the representation up-to-date by automatically capturing and incorporating any changes (Dimyadi and Amor, 2013).

- **Class 3: DMC for specific domains.**

Objective:

Studies are often associated with an explicit domain, such as spatial assessment, structural integrity, safety, energy usage, and so on. This means that the purpose of the DMC process will define the objective of the research and the monitored criteria to accomplish that purpose.

Limitations:

As long as followed criteria for this class assists only the designated output, it may be hard to generalize these methodologies on other applications. Furthermore, there has been a lack of using the ontological foundations of the checking process presented in Class 1, therefore there are always unanswered questions about why does the author choose this approach? , and is it the best approach to be followed?

- **Class 4: DMC across project phases.**

Objective:

Conducting the checking process while considering the project phase is the main characteristic of this class. Furthermore, it defines the mandatory information for a defined type of object for a defined stage and role in the building process. Exchanged information models throughout the project phases are modified and supplemented taking into account the essentials of the new activity, and the continuance of the information modeling process (Galkina and Kuzina, 2018).

Limitations:

The lack of defined information in the digital model limits the utilization of information. The limited content of information is connected with the lack of agreement about what type of information should be exchanged. This agreement may require a formal framework to remind involved parties of the core tasks that should be undertaken at any particular stage, and accordingly well-defined exchanged information.

Generally, there has been a lack of explicit research that proposes a framework consolidating the entire process of Model Checking. Researches on classes 1 to 4 may compose the integrated image of Digital Model checking only if it has been read together and in conjunction with a solid framework.

Hjelseth (2015) offers a broad approach, ranging from exploring principles of model checking to practices in state-of-the-art companies, in addition to reviewing commercial software. This approach could be considered as one of the most comprehensive models although it does not comprise the whole image that embraces the practitioner's contribution, supplementary applications, and corresponding project phases.

Furthermore, Hjelseth (2015) proposed a unique classification of DMC maturity levels for compliance and content checking through the Five-level DMC maturity model as shown in Figure 4. An overview of the five levels and the corresponding description is presented as follows: 1) Level 5: Pervasive model checking, 2) Level 4: Integrated model checking, 3) Level 3: Specific purpose checking 4) Level 2: Adjusted model checking and Level 1: Clash detection checking. Promoting these levels within the AEC industry may allow involved stakeholders to determine what and how they are willing to achieve their aims. Nonetheless, a consistent context, proper adaptation, and classification aiding DMC are still missing, which is very important for further studies.

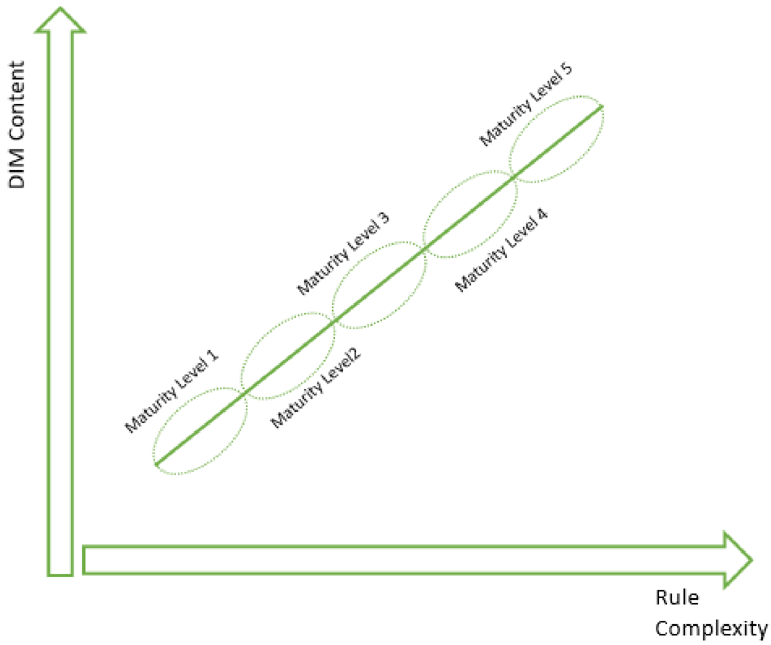

Figure 4 - DMC Maturity levels

2.3. Timeline of Digital Model Checking

In recent years various efforts were undertaken in order to develop a method for Digital Model Checking. Historical developments have been commenced by exploring techniques that turn thoughts and textual regulations into the format required for information processing. These techniques could be named "Encoding". Accordingly, several organizations and software developers have been introducing kinds of systems that advance the processing of information thru checking platforms. The timeline of encoding and checking platform is briefed as follows:

2.3.1. Encoding Prescriptive Regulations

The importance of DMC was initiated in line with the interest of converting building codes into a format suitable for machine interpretation and application. Encoding efforts are presented in Figure 5 when Fenves made the observation that decision tables, if-then-novel programming and program documentation techniques could be used to represent design standard provisions.

Afterward and throughout the 1980s, a system was developed, called the SASE (Standards Analysis, Synthesis, and Expression) (Wright and Reed, 1987) to provide a comprehensive structure for families of related codes (as exists across the many US jurisdictions). Besides, the use of Hypertext Markup language (HTML) to represent textual regulatory rules was emerged (Dimyadi and Amor, 2013). To overcome the limitations of HTML, Extensible Markup Language (XML) was designed and been widely used throughout the 1990s. The investigations into an automated or semi-automated interpretation of information from regulatory texts into machine-readable rules using Natural Language Processing (NLP) techniques have continued until today (Cheng et al., 2008; Zhang and El-Gohary, 2011).

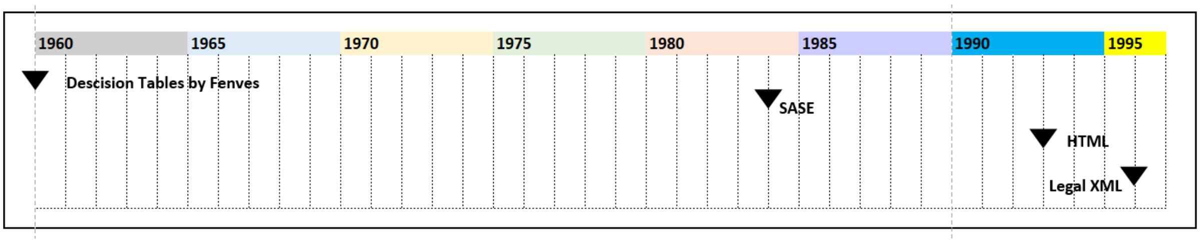

Figure 5 -Timeline of Encoding

2.3.2. Checking Platforms

A brief of common checking platforms with a timeline is exemplified hereunder, Figure 6:

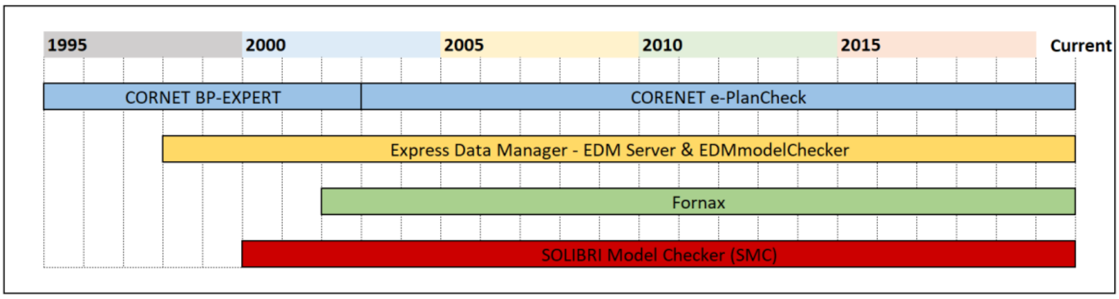

Figure 6 -Timeline of Checking Platforms

CORENET

In 1995, the Singaporean Building Construction Authority (BCA) introduced the CORENET (Construction and Real Estate Network) (BCA Singapore, 2013) to re-engineer and streamline the fragmented work process in the construction industry incorporating an internal developed System to check 2D plans for compliance. The system was upgraded in 2002 to CORENET e-Plan Check replacing the 2D System with the 3D IFC data model (Khemlani, 2005). The provided checking functionalities focus on the national applicable codes in the areas of building control, barrier-free access and fire safety.

At this point, it should also be stated that the introduction of CORENET was seriously promoted by the Singaporean government and accompanied by appropriate legislation. As a consequence, Using CORENET has become obligatory to receive a building permit for construction projects. However, it remains doubtful that such an approach would prevail in the same way in the Middle East or European countries (Preidel and Borrmann, 2015).

Express Data Manager

The Express Data Manager (EDM) platform was established by Jotne EPM Technology in Norway in 1998 as an object database with tools to manage complex product data models. It started as a collaboration tool but has since incorporated several additional modules including EDMmodelChecker that supports open development using the EXPRESS data modeling language (Yang, 2003). EDM Model Checker can be also utilized to validate a data set and ensures that it conforms to all rules and constraints defined in one or more EXPRESS Schemata (Nawari, 2018).

Further, EXPRESS Data Manager (EDM) offers functionality for data exchange, data sharing, data integration and data archival that may be used to build data translators/converters from one data format to another one, where one of them may be, but does not need to be an international standard, such as STEP or PLCS.

FORNAX Library

FORNAX is a C++ library that has been established in 2002 that donates regulatory rules whether as an independent regulatory data representation directly or via other dependent systems. For example, CORENET e-Plan is using the Fornax library in conjunction with EDM that has regulations and additional rules hard-coded in EXPRESS.

Solibri Model Checker (SMC)

Solibri Model Checker (SMC) was developed in Finland in 2000 and started as a BIM model quality assurance and validation tool, but has since developed into a stand-alone graphically-driven rule-based compliance checking and reporting application. SMC has a set of built-in rules that can be managed by a ruleset manager. A ruleset can be replicated, but the extent of user customization is limited to changing parameters (Eastman et al., 2009).

3. MODULARIZATION APPROACH FOR DMC

To achieve the objective of this research, this study proposes a diagnostic approach based on using pre-defined characteristics to analyze digital model checking (DMC) and propose a formal framework and implementation plan. The pre-defined characteristics are the DMC principles that have been illustrated previously in section 2.1. In other words, the modularization approach focuses on "what things are", "what is the logic behind extending the pre-existing concepts" and "how it assists the DMC process". These modules play a fundamental role in a way that they must be captured, tracked, and interconnected during the development of the research framework and the implementation plan. The goal of this modularization is to initiate a framework that concludes the most important features to build a DMC concepts and then gather these features into an implementation plan which should be executable and leveraging the DMC process.

The arrangement of modules in Figure 7 schematically demonstrates the use of the model checking principles as a solid foundation that has been expanded to capture other layers of DMC regardless of the nature of the digital model, complexity of the rule and the system solution provided, therefore Figure 7 indicates the three modules of the framework and the corresponding layers lie beneath each module.

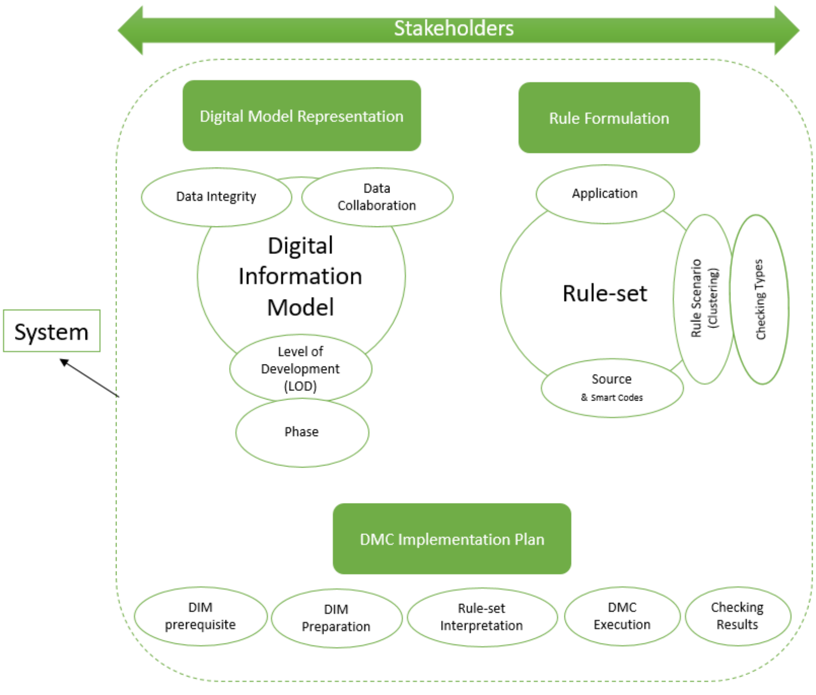

Figure 7 - Three Modules of the proposed framework for DMC

In summary, the modularization of this research has proposed a conceptual DMC framework that can be structured into three modules named as follows: 1) Digital Model Representation, 2) Rule Formulation, and 3) DMC implementation plan. To simplify, code compliance check, as an example, is considered as an application that is common and desirable has to be *digitally represented* while taking into consideration the *formulation of the associated rule-sets*, classification of applied rules, matching checking types and ultimately *incorporated into a proper DMC implementation plan*. The modules are detailed as follows:

First Module

This module is concerned about the digital model representation that has been designed utilizing the digital information model (DIM) as one of DMC's basic principles. The representation of DIMs may vary based on numerous factors such as the scale of a project, project phases, role of stakeholders, local regulations, organizational policies, etc. The common request among DMC’s performers is to ensure that an appropriate level of information management and quality is in-place. Therefore, the basic concept of the proposed module is to evaluate the competence of DIMs to *integrate*, *exchange*, *identify*, and *verify* its content in a way that supports the best performance of DMC.

The answer to the question of “what things are” and the conversion of the concepts has led to the need for three layers that are a) Data integrity b) Data Collaboration c) Level of development and Project phase. These layers purpose to mitigate most of the digital model defects generated in unstructured, unmanaged, unused and immature information that cause the fundamental problem of data utilization. The logic and the influence of these layers on DIMs and accordingly the DMC process is as follows:

a) Data Integrity

The magnitude of data integrity is essentially a viable accumulation that fulfills the proficiency of the DIMs. Data Integrity solidifies the information management strategy of the DIMs by assessing its competence and truthfulness. The importance of data integrity has arisen due to its nature of not only fixing mistakes but also it ensures expressive inputs for the DMC means. In other words, data integrity ensures adequacy, recoverability and searchability, traceability, and connectivity. Integrity derived logic is merely condensed as “output will only be as good as the inputs”. Therefore, data integrity aids the DMC process by offering meaningful input.

b) Data Collaboration

In essence, DMC is a kind of process that mandates the participation of several stakeholders providing that DIMs and checking results should continuously be available. This should also occur via an interoperable format to protect DIMs and avoid data deficiencies. The logic derived in this layer aiming to ensure a successful data exchange of interoperable DIM’s formats within a shared environment where entitles are rather to work openly together contained by a collaborative environment. By achieving that, data collaboration will be supporting the DMC process with a successful exchange of information along different stages of the project life cycle where the DMC performers will be able to access the DIMs and share results with other concerned stakeholders.

c) Level of development (LOD) and Project Phases

DIM content is a set of data that has to be identified based on a formal specification in order to be later verified. LOD is similarly an essential entity that considers as a specification to articulate with clarity the content and reliability of the information at different levels of development (BIMForum, 2019). It is worth mentioning that a relationship between **LOD** and **project phases** can be established. The phases of the project lifecycle have different LODs, which identify the structure and detail that rules can apply meaningfully (Solihin and Eastman, 2015).

Second Module

The emergence of the second module commences similar to the way in which the first module has been established but by using the rule-set as a centerpiece. This module concerns the formulation of rules imitating four important inquiries:

a) The possible causes of a rule and the prescribed source that gather these rules into a formalized rule-set. This inquiry is being defined over a layer named "**Source of rules".**

b) The uses of a rule-set and corresponding performances in the AEC industry. This layer is concluded by a term named "**Applications"** (e.g., Clash detection, code validation).

c) The characteristics of formed rule-set and sequences of applying this rules-set exploit the necessity of having a formalized classification of rules entitled **Rule-scenarios** or **Rule-Clustering.** This classification entails potential circumstances dependent on rule definition and requirements, typical use of the rule, explanation of the complexity, and suitability of utilized techniques (Solihin and Eastman, 2015).

d) Eventually, distinguishing the optimal solutions to pursuing DMC applications (the transformation of inputs into results). This activity is achieved by capturing **Checking types**.

The main characteristic of this module is the dynamicity that evolves constantly progresses, modifications, and expansions to satisfy the requirements and future targets of the (AEC) industry. Therefore, the discussed layers will probably provide a substantial breakdown that eases confronting any potential risks.

In essence, the layers discussed above are not independent which means that censoriously evaluating the interconnection is mandatory. This is the reason that this module will investigate the case-effect of the related layers such as "Application VS source of Rules" relationship and "rule scenarios VS checking types" that conceivably will ultimate the demand for finding the equivalent approach that decodes rule scenarios into authenticity.

The cause-effect of this module is not limited to its associated layers but it may correspondingly influence the layers of the first module. Therefore, it is vital to mention that the objectives of the second module are not restricted to ensure a safe formulation of a rule-set but also it includes a list of influences on the representation of the DIMs.

Having the scope and source defined (first and second layers), rule classes (third layer), and control on the transformation process (fourth layer) is vital to all involved stakeholders along a building life cycle. Not also practitioners but also scholar's attention for future research and development where the industry will benefit most as well (Solihin and Eastman, 2015).

Third Module

This module is concerned more in materializing first and second modules into a feasible plan that has to be charted to ensure a successful DMC process. Analyzing the characteristics and obligations of first and second modules will probably justify the criteria of leveraging DMC by merging the knowledge of practitioners and scholars in an implementation plan. Nonetheless, this module will not add new concepts, but it will provide a roadmap that connects the layers illustrated previously. In other words, the third module designates the documented steps needed to complete DMC implementation activities and ensure all outcomes are achieved.

Effective DMC Implementation plan mandates that the stages described in this module primarily include initiation, preparation, interpretation, execution, and ultimately proper closure. Accordingly, the implementation plan has included the following layers:

a) DIM prerequisites

This layer purposes to outline the preconditions of a DMC process that commences with the readiness of the DIMs. These conditions may include substances related to the role of involved stakeholders along DMC stages, the production and exchange measures of DIMs, the requirements of rule-set translation into an executable technique and nesting systems.

b) Rule interpretation

Rules in legal documentations are in an original textual form that is not suitable for implementation in a DMC system. It is required to be formalized as a machine-readable rule to be presented in the form of logical expressions.

c) DMC Execution

This layer launches when the DMC type is recognized, DIMs are well prepared, rule-sets are translated properly, and the nesting system is selected and ready for practice.

d) Checking results

The final station of the DMC process that declares states of the resultants. The output of this layer may either announce officially close-out of the DMC process or mandate updates on the DIMs. This happens in accordance with the data feedback mechanism that ensures the feedback loop will effectively close-out the DMC process.

3.1. Systems & Stakeholders

The three modules are influenced and contained continuously by an operating mechanism that is called "**System**". Each module may have to be executed within a mechanism or interconnecting network that allows for an organized schema to establish all work together in a nesting **system**.

The current approaches for performing DMC or its sub-processes thru Systems are either based on 1) Proprietary framework, 2) Domain-Specific Area or 3) Hard-Coded Rule-Based. Examples of systems are given in Table 2.

Table 2 - Example of Systems

System	Examples
Proprietary framework	*Revit *SMC (Solibri, 2016)
Domain-Specific Area	*XML – data encoding *SQL – databases *CORNET (web-based)
Hard-Coded Rule-Based	*"BIM Interoperability Plugin" by Autodesk

Systems do not share the same consistent role along with Modules. To illustrate, the system responsible for authorizing the DIMs, which forms the core of the first module, is not obligatory to be the same system used for forming and executing a rule-set.

An important feature that should be taken into consideration while evaluating systems is the "Interoperability of Systems". Whether the systems are federated, unified, or integrated, it should assure certain level interoperability that could be achieved through openness, machine readability, agreeing with industry standards, and proper standardized specifications. In this sense, interoperability between DIMs authoring platforms and DMC tools remains a major issue.

On the other hand, assigning an accountable driver of each module is crucial. The wider illustration of this driver is a "**Stakeholder"** who has a different role according to the phase of involvement**.** Within any process, stakeholders and associated roles may add or eliminate requirements or constraints based on information from their specialty. However, this will be significant to have when understanding project constraints and risks. The more engagement and involvement of stakeholders, the more containable and reduced risks disturbing the (AEC) industry.

Theoretically, this research assumed that the authorization of DIMs and associated identifications such as project phase and level of development are achieved by stakeholders (Driver 01). The formulation of rule-sets from various sources may also necessitate another stakeholder (Driver 02). Implementation of DMC may be accomplished by one of the previously identified stakeholders or by a newly involved Stakeholder (Driver 03). Within each driver, there are many different roles and responsibilities consistent with the sub-processes.

4. FRAMEWORK DEVELOPMENT

The contribution of this research is in dealing with the DMC process from a different standpoint that evolves the strength of DIMs, cause-effect of rules, project phases, systems and involved stakeholders. Furthermore, DMC is a kind of process that has former and subsequent project activities. Therefore, the proposed development will also enable the interaction between various stakeholders and processes lengthways the project life cycle irrespective of any difficulties similar to the diversity and complexity of given rules and scale of the project. For that reason, this chapter will commence with detailing the predefined DMC modules and emphasize the relations between the layers in order to accumulate the gained knowledge into a practicable implementation plan that guarantees the success of the proposed framework.

4.1. Digital Model Representation

Digital models are massive datasets, even for medium-scale constructions. Therefore, there are two major parts that digital models deal with. The first is the Information Integrity & Collaboration, and the second is the project phase and level of development (LOD). However, it is noticeable that almost the major studies of DMC have not investigated the effect of data integrity and collaboration in the DMC environment assuming it is approximately taken for granted. In this sense, this section will explore the following:

Data Integrity

This layer is a critical concept for authoring the DIMs, design of a DMC process and implementation stages. Maintaining data integrity means making sure the data remains accurate and consistent over its entire life-cycle (database.guide, 2016). This means that data integrity is a prerequisite to the DMC implementation plan that could successfully engage DIMs into a DMC process.

Data integrity is also a vital milestone for authoring the DIMs. The performance indicator of this milestone has to be guaranteed prior to the start of the DMC process. Based on that, DIM authors and DMC performers should agree on the required performance of data integrity that allows for a smooth hand-over process between the DIM author and DMC performer. It is worth mentioning that DIM authors and DMC performers could whether to be the same stakeholders or two independent entities. However, in both cases, data integrity has to be leveraged.

This research merely investigates the influence of data integrity on a single DIM (Intra-disciplinary) and several single DIMs (Interdisciplinary) where many scenarios could place data integrity of DIMs at risk of being corrupted, inaccurate, or inconsistent. Less reliability among DIMs means a weak process that could not elaborate meaningfully. This research intends to supplement the DIMs by offering some types of integrity that have to be existent before the DMC kick-off. Compliance with these integrity types returns with DIMs ready for further use in a DMC process. These types are presented as follows:

Integrity of Inputs

DIMs are authored using textual, graphical or tabular information as inputs that have to be digitally transformed. Examples of inputs may embrace 2D drawings, sketches, specifications, client requirements, regulation, conceptual briefs, etc. If the inputs are not integrated, this means that DIMs may not be digitally represented providing that DIM authors should expect many obstacles and several gaps throughout the modeling process. Even if the authors successfully generated a model, that model doesn't characterize the original objective and rather be not used furtherly. That case could be clearly illustrated when, for example, columns positions in 2D Architecture drawings are not in the same positions of the 2D structural drawings. Another example is while transferring data between two databases, the designer accidentally inserts the data into the wrong table. Therefore, the integrity of inputs is a central cornerstone for the authoring of DIMs where healthy-data, data-match, and data-structure should encounter.

Throughout the life span of a project, DIMs usually collect inputs thru different project phases that could be at early design stages or the end of a process as of the facility management. However, the integrity of inputs should capture, but not fully, other integrity types such as integrity of geometry, relations, etc. that will be presented in the upcoming paragraphs. This is to say that integrity of inputs is concerned about the totality of information and any possible relation or interaction.

Integrity of Geometry (objects)

This type of integrity is concerned about the geometrical characteristics of a model element. In other words, the integrity of geometry ensures that predefined constraints of any kind of geometry are dominated. For example, the definition of a 2D rectangular shape is four axes that meet each other with right angles, thus the integrity of the geometry occurs only if these conditions have been accomplished. Another advanced application is exemplified whereas modeling a roof of a building that could be either modeled on Revit (Autodesk, 2020b) using "Floor" or "Roof" categories. Both categories can draw a roof, but "Roof" will add consistency and accuracy to the geometry. Similarly, the geometry of any DIM's model elements should be complying with some constraints to express the way model elements should be digitally represented and integrated.

Integrity of attributes

Attributes are the information that determines the properties of a model element, tag in a database, or a string of characters. Each model element has its attributes that could be specific for that element or shared along with other model elements. For instance, "number of risers" can only be a property associated with stairs, but not for a structural column while "structural material" can be a common property between stairs and columns. Nevertheless, the integrity of attributes seeks consistency between model elements that ensures accurate and consistent exchange and extraction of information.

The extent of the integrity of attributes should not only include a group of model elements in the same DIM since this type of integrity is desirable along with federated (several single models) and coordination (several linked models) DIMs, but also laterally various stakeholders.

DMC rules usually encompass constraints to the attributes, therefore there is a strong connection between rule-sets and the attributes of DIMs in which attributes have to be well-defined. In this manner, the standardized data model by BuildingSmart, IFC schema (BuildingSMART, 2020), compromises a rational method to ascertain the characteristics and attributes of elements that can elevate the integrity of attributes. This also emphasizes the continuous use of IFC among DMC platforms.

Integrity of Values

The synchronized integrity type to the attributes is the corresponding value that has to be precise, within an acceptable range and using the correct format. The Fire-rating property of a wall, as an example, may have a value of one or two hours. If the value is two minutes, this means imprecise and unacceptable range. Similarly, if the value is two kilometres per hour, this means an incorrect format. Thus, it is possible to claim that the integrity of values could be easily controlled if a system could prohibit unintegrated values; however, this research would rather communicate that the integrity of values should be always monitored by users.

Integrity of Relations and Constraints

DIMs are authored based on a number of constraints and relations that give a building system kind of applicability into reality. “Window must be nested within a wall” is a relation and constraint at the same time. The relation is that the window has to be hosted by a wall, and the constraint is that the window dimensions should be contained by the wall boundaries. Integrity of relations and constraints enforce data integrity by the means of constraining or restricting the data that users can model, insert, delete, or update in the DIMs.

Furthermore, the integrity of relations and constraints is dependent on the integrity of geometry, attributes and values. This could illustrate the importance of these types of integrity to the relations and constraints.

It is also worth mentioning that smart systems and authoring environments should be able to partially detect deficiencies of relations and constraints in addition to the involvement of the users. So far, authoring environments offer the integrity of relations and constraints among elements of federated DIMs, but not coordination DIMs unless it becomes a rule-set. To illustrate, the previously mentioned example of “Window must be nested within a wall” deems as a rule that could be performed to coordination DIMs using DMC systems.

4.1.1. Data Collaboration

DMC is a process that has to be performed several times along a project lifecycle, and each time of the DMC process may be also gone through numerous iterations. This involves different stakeholders to work together or separately based on their specialty and the operation scale, phase, type, or time. This means that shared data sources have to be visualized by the right stakeholders in the right format and at the right time for the sake of effective decisions as shown in Figure 8, and as follows:

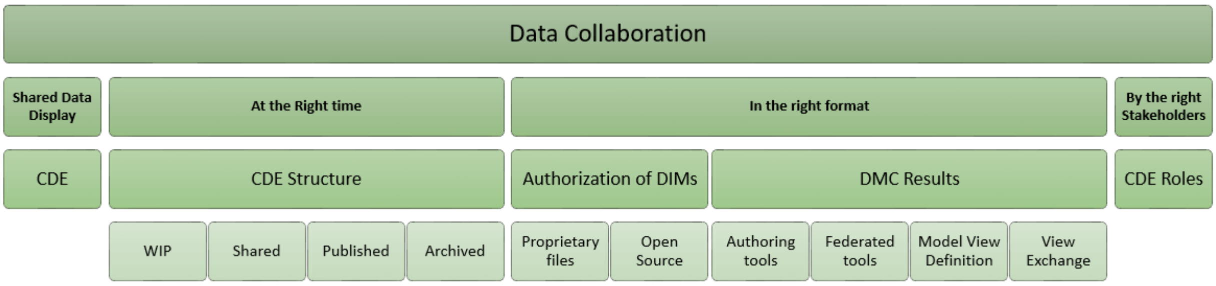

Figure 8 - Data Collaboration

Shared Data Display

Data collaboration should guarantee an interactive display that allows DMC performers or other stakeholders to use the DIMs in a DMC process and share the output for further corrective actions. The equivalency of interactive display in digitalization is a Common Data Environment (CDE). As per ISO19650-1:2018 , CDE is "*Agreed source of information for any given project or asset, for collecting, managing and disseminating each information container through a managed process*".

At the right Time

The timing of sharing DIMs is a key factor for an effective DMC process. Clash detection tests, for example, should not be performed unless coordination models are released formally by their authors, otherwise, the clash test report will show considerable false clashes. This concern has been covered by the structure of CDE that establishes four states of DIMs according to their readiness for being shared and communicated for new model uses and properly with other stakeholders (ISO19650-1:2018).

- **Work in progress:** used to hold unapproved information, and cannot be shared.
- **SHARED**: contains checked, reviewed, and approved info for sharing with other stakeholders. This considers the optimal sharing state of DIMs that guarantees the ultimate quality of the project deliverables.
- **PUBLISHED**: authorized information that is shared for new models uses and possibly with other stakeholders. For that reason, ensuring an accurate closure of the DMC process is mandatory before hand-over with other stakeholders. This means that DMC's corrective actions may not be all implemented, by way of it may be just enough to attach the checking results with the PUBLISHED DIMs without updating the DIMs.
- **ARCHIVE:** records activities of each project delivery milestone that may also include the corrective actions by a DMC process.

In the right Format

This concerns the possible formats of both the DIMs, as inputs, and DMC results, as output. Starting with the **authorization of DIMs** format, as the **first option**, it is either **proprietary files** that are encoded secretly and only can be used through a proprietary system or it is **open source** that allows users to encode or use without any constraints and this is named non-proprietary files.

Second is the DMC results formats that could have four options. The first option is when the results are part of the **authoring tool** such as the "interference check" in Revit. In that case, the results format will follow the authoring tool, and both will have the same file format.

The second option is when using **federated tools** such as SMC or Navisworks (Autodesk, 2020a) providing that these tools cannot author or update DIMs, but it just performs some sort of DMC and then presents results.

Federated tools may also offer a mapping feature to enable the users to select and zoom-in on the flawed components in the DIM's authoring tool. This feature has been found in many DMC systems such as the "SwitchBack" function between Navisworks and Revit.

The file formats of the federated tools are dependent on the possible export feature of the federated tool. For example, Navisworks, by Autodesk, does not automatically export their proprietary formats (e.g. NWC or NWD) into IFC format unless a hardcoded plugin, named "IFC exporter" is installed. On the other side, SMC provides more flexibility in export the results by interoperable open-source formats such as XML, IFC, and BCF (BuildingSMART, 2009).

The **third option** is limited to a sort of DMCs that outcomes with subdivisions of DIM's complete data. These subsets are named **Model view definitions** (MVD) controlled by a documented guidebook that is named information delivery manual (IM). MVD does not have a unique format as it follows the format of the original DIMs.

The **fourth option** is **view exchange** that merely visualizes and locates the issues within the DIMs via designed views in order to be shared among stakeholders. The given example of view exchange is an XML schema that is called BCF that encodes messages to report issues found within DIMs. This will save the implications of sharing the entire DIMs and will enable a degree of collaboration.

By the right Stakeholder

CDE, as a common platform for involved stakeholders, has no specific constraint about the role and responsibility of each stakeholder. This is normally organized by a set of limitations and permissions that may be outlined in BEP or internally within a working project team.

In this respect, Autodesk thru BIM 360 (Mekawy and Petzold, 2018) has offered theoretically three project roles within a CDE that are Viewers, Editors, and Project Administrators. These roles may exist with any stakeholders disregarding their scope or associated project phase. For example, contractors in their organizational hierarchy have Viewer, Editors, and Admins, so do designers, owners, and facility management firms. Each role that a list of tasks that cannot be done by other entities, and even this task list has internal agreements that organize the work between the team members.

However, this research still considers that the given example is not sufficient to represent the roles and responsibilities of stakeholders alongside the project life cycle in the presence of a CDE. To illustrate, trying to match the DMC performers with the three given roles is not possible since DMC performers are not only viewers, may do some editing, and they are not admins in most of the cases. Adding a new role that is called "Reviewers" may seem a good solution to this explained gap. The reviewers' task list has to adopt the nature of their responsibilities that should include view, controlled access, preconditioned edit and share.

4.1.2. Level of Development (LOD) and Project Phases

When there was no datum in the world for DIMs, an analogy had to be found for reflecting the concept of geometric resolution and degree of elaboration (Borrmann *et al.*, 2018). The qualitative characteristic of a digital information model (DIM) is an important factor affecting the building environment that has led to the establishment of the term “Level of development” in American institutes or “Level of detail” as commonly known in the UK.

American Institute of Architects (AIA) revealed the term ‘levels of development’ in 2008 defined as the degree to which the element’s geometry and attached information has been reasoned. Advanced published Level of Development Specification was introduced by BIM Forum in 2013 based on AIA protocols. This classification has 6 levels (100 – 500). In contrast, the term Level of detail was introduced within the protocol released by AEC (UK) in 2009 that was considered misrepresentative as it sets much emphasis on the geometric appearance (Borrmann *et al.*, 2018).

Later in 2013, PAS 1192-2 (PAS 1192-2, 2013) Specification introduced ‘Level of Definition’ as a new classification system with seven levels (1-7) to include both aspects of ‘Level of Model Detail’ (LOD) and Level of Model Information’ (LOI) (Level of Definition = LOD + LOI). Although LOD may be illuminated in dissimilar methods depending on where in the world, nonetheless, the basic concept remains the same within the digitalization process that LOD would represent the graphical and semantic maturity of DIMs. In this paper, the abbreviation LOD stands for the Level of Development, which represents the composition of both Level of Geometry (LOG) and Level of Information (LOI) (semantics).

The more DIMs are becoming more complex and detailed, the more need to LOD to ensure that information is well-developed for the sake of collaborative integration and typically sharing organized and standardized information within the DMC process. Not only the LOD but also the corresponding project phase are key factors that guide us in organizing DMC by the next ideas:

a) The potential use of DMC mandates that DIMs should share the same LOD in a specific phase. To illustrate, multidisciplinary coordination requires that DIMs from different disciplines are at the same level of development (LOD) or in the same project phase to perform DMC efficiently (Hjelseth, 2015) otherwise the rule-set may be inappropriate and respectively the results may be misleading.

b) Prerequisites and requirements of a rule-set may influence the associated LOD where a rule may require additional geometrical or semantical information to the DIM. For instance, any rule that concerns about concrete reinforcement mandates that the element itself should have advanced LOD (e.g. LOD 350) according to (BIMForum, 2019).

c) LOD is a term that is interrelated to project phases and other consistent processes derived by stakeholders who are either responsible for authoring DIMs or performing DMC. As far as of today, there is no approach for formally linking multiple levels of development throughout the life span of a project. Neither is there no formal definition of a project phase that defines the LOD of a building component nor is there an explicit description of the fluffiness of its geometric and semantic information. For that reason, Table 3 presents an example of the schematic relationship between UK&US LODs, project phase from the 'RIBA-Plan of Work', 2020, and stakeholders as in Uniclass 2015_Role (National Building Specification, 2015). The hypothesis of this table is to exemplify the importance of the LOD and corresponding project phase to the digital model representation. Moreover, it is necessary to determine which stakeholders should receive which information at what level of development at a certain moment in the process that why Table 3 has also included the possible role of each stakeholder at each phase through drivers of each stage. The table is endorsing the use of benchmarked classification in effect to enforce the cause-effect on the DMC process. It is also vital to mention that the future practices of the given example have to be related to national local codes and best practices.

Table 3 - LOD, Project phase and Role of stakeholders

UK LOD	US LOD	Description	Content	Stakeholder (Role)
1	LOD 100	Brief	*Client requirements and Site constrains. DIMs would by symbols and non-graphical information. *Equivalent to **RIBA stage 0&1**.	***Ro_10**: Management ***Ro_50**: Design
2	LOD 100	Concept	*Conceptual or Massing model including basic information. * Equivalent to **RIBA stage 2**.	***Ro_10**: Management ***Ro_50**: Design
3	LOD 200	Developed Design	*Project objectives are defined and more informative model. * Equivalent to **RIBA stage 3**.	***Ro_10**: Management ***Ro_50**: Design
4	LOD 300&350	Production	*Pre-construction and end of the design stage. *Model Elements are accurate and coordinated. * Equivalent to **RIBA stage 4**.	***Ro_10**: Management ***Ro_30**: Delivery ***Ro_70**: Site
5	LOD 400	Installation	*Model is suitable for fabrication and assembly with construction requirements and specific components. * Equivalent to **RIBA stage 5**.	***Ro_10**: Management ***Ro_30**: Delivery ***Ro_70**: Site
6	LOD 500	As-Built	*Model is suitable for handover. *Showing the project as it has been constructed. * Equivalent to **RIBA stage 6**.	***Ro_10**: Management ***Ro_30**: Delivery ***Ro_70**: Site
7		In Use	* Model is suitable for the facility, operation and asset management. * Equivalent to **RIBA stage 7**.	***Ro_10**: Management ***Ro_30**: Delivery ***Ro_70**: Site roles

In order to properly integrate LOD and project phases into the DMC process, it possibly will require additional developments to DIMs, to be mapped with rule-sets or unified during the execution of DMC; though, these influences have to be strategically documented and properly implemented into the DMC life cycle.

Although LOD requirements and project phases are typically part of the Exchange Information Requirements (EIR) defined by the client at the beginning of the project, the LODs are commonly defined with respect to the uses of the DIM disregarding the influence of the DMC.

LODs may also consider as a source of rules by the means of initiating a kind of checking that validates whether the elements of the DIMs are compliant with predefined LOD or not. An example of this given case is provided when a rule-set has been implemented into a check named “Required LOD” offered by SMC that offers LODs compliance checks.

Since the requirements of DMC may work symphonically with schemes at varied LOD, it is highly recommended that LOD should be at the lowest level adequate for DMC (Solihin and Eastman, 2015) to avoid the possibility of having unnecessarily big data that causes deficiencies in production and competence.

Winding up this topic to emphasize on the fact that without a proper LOD definition in place, DIMs can turn out to be less significant for the entire involved stakeholders. With proper Level of Development specifications, the accuracy, precision, and value of the entire DMC process, as well as DIMs, can be enhanced phenomenally for the entire lifecycle of a project. Furthermore, LOD along with DMC uses should be a key aspect in project delivery documentation prior to the kick-off the major milestones and key performable indicators (KPIs).

LOD creates a standardized definition of what completion means and eliminates chances of discrepancies associated with project completion. Using LOD, parties working under different disciplines can communicate with each other in a better way with greater clarity. LOD enhances clarity in a process by making use of advanced techniques and technology.

Last but not least and based on these understandings, it is envisaged that the role of the LOD and Project phase will serve actively as a decision support system for the DMC process and the organization of model-based collaboration.

4.2. Rule Formulation

Rule-sets for the AEC industry are a prescribed guide for specific action or conduct. This section merely focuses on 1) how rules are generated 2) what the main characteristics of each source of rules are and 3) what the possible applications are. Once rules are used appropriately, it provides a sense of predictability and consistency for DMC, thereby promoting the formulation of rules that helps guide actions toward desired applications.

4.2.1. Source of Rule-sets

A rule-set is a group of rules for a specific purpose that has to be sourced, so stakeholders will become able to critically evaluate the relevance of these sources prior to relying on the inner rules and ensure that DIMs have got the necessary properties and attributes to achieve the objective of the DMC.

In this context, this section will try to minimize the existence of unsourced rule-sets where each source of rule-sets will be critically evaluated in terms of complexity, appropriateness and reusability. The incremental modifications to rule-sets can be made by expressing exceptions to existing rules rather than reengineering the entire set.

The source of a rule could be formally acknowledged as an accountable source according to 1) the ability to be applied across the different building environments, 2) reusability across various stakeholders such as public agencies, organizations or clients and 3) where applications under assessment are commonly used, such as rules associated with fire and life safety, structural analysis, energy usage and costs (Eastman *et al.*, 2009). Therefore, there is a high possibility that rule-sets fall into one of the following sources as shown on. Figure 9:

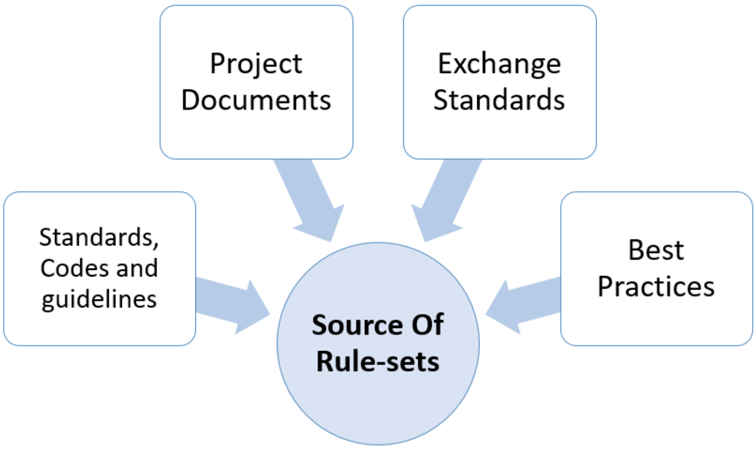

Figure 9 - Sources of Rule-set

1. **Standards, Codes and Guidelines** are a significant source of rules that could include several compulsory rules to assure safe and high competent building environments. The authors of this source are mostly scholars, researchers, non-governmental agencies and public institutes who contribute to society by forming regulatory conditions. Subsequently, these conditions are subject

to be formalized by governments, public agencies, or organizations as an obligatory procedure prior to issuance of certificates, building permits, licenses or any sort of authorities approvals.

The automation of DMC sources has generated **SmartCodes** that represents the digital representation of the provisions and regulations of codes and standards (Nawari, 2012). SmartCode provides the legislative form with an authoring environment to manage the alterations of this kind of source (Dimyadi and Amor, 2013). The wide use of SmartCode depends on the adoption of local authorities and organizations where the system should be able to handle big data environments and also will probably require a high budget, strict control, constant modernizations in order to be publicized for commercial use and utilized efficiently.

The main characteristic of this source is the steadiness along a project's life cycle and the low change frequency on the constitutional level. For instance, it is not common that a code of practice might change throughout a running project, and in most cases, it has never been obligatory to follow the latest version in a middle of a process. In practice, this source encompasses a high level of predictability and traceability. Knowing the characteristics of this source is important to software developers and interpreters who encode and implement the rule-sets into systems and checking platforms. Furthermore, the complexity of the rule-sets outcomes from this source is relatively high. One example form ADA Accessibility Guidelines for Buildings and facilities for stairs height described is:

a) Stairs are required to be ADA accessible, and
b) To have uniform riser heights, Uniform tread depths, and
c) Stair risers must be a minimum of 4 inches tall, Cannot exceed 7 inches tall, and
d) Treads (the horizontal surfaces) must be at least 11 inches in depth, and
e) Tread surfaces (from side to side) cannot change in level and treads cannot have a slope greater than 1:48, and
f) the nosing (or leading edge) of a tread must be a maximum of 1/2 inch.

This rule requires more complicated logic and concepts in interpretation and execution.

2. **Project Delivery Documents** serve as “contractual agreement” between various involved stakeholders to assure proper information delivery of a building environment. According to BSI_PAS_1192_2 (PAS 1192-2, 2013), project delivery documents should contain major project milestones, applications, uses, and requirements related to the delivery and exchange of DIMs that could be translated into rule-sets. However, checking for compliance with delivery documents, not only the milestones, is an additional source of rule-sets that endorses a successful delivery process. In other words, the content of project documents itself could be considered as a source of rules or it may direct to other sources. For illustration, a post-contract BIM execution plan (BEP) usually comprises uses such as clash detection and code validation. Clash detection is a direct application, but code validation will lead to the first source of rules “Standards, Codes and Guidelines”. This emphasized the looping effect where the rule applications are similarly a source of rules-sets either. The similarities in these types of rule-sets are limited due to the numerous variables in position. The variables affecting this kind of rules are unpredictable and influenced by several parameters as

project nature, the guidance of local authorities, type of contract, involved stakeholders, assigned roles, project phase, and process timing.

Although generated rules of this source seem common, it is subject to prompt progression and alterations that lack predictability and traceability. The reasonable conclusion of these individualities results in concerns about 1) required domain knowledge and 2) competences of systems conducting DMC to deal with this wide range of variables in a timely manner and 3) looping effect and continuous updates to DIMs.

3. **Best Practices** are a set of rules or notions that represent the most efficient or judicious sequence of action, in a given process. Best practices may be established by authorities, syndicates, or they may be commanded by an organization internally. The high degree of reusability of best practices may lead to a kind of escalation that directives best practices to become legalized and standardized.

   One of the most mutual best practices that affect DIMs is rule-sets associated with modeling methodology where some prerequisites are required to achieve a certain application of DMC. For example, dividing staircases by stories while modeling and performing clash detection is compulsory to know at which story the clash has been positioned otherwise the clash report will display misrepresentative results. The rules of this source are privileged with being simple with limited complexity.

   Further best practices may come around such as checks for duplicated model elements, project coordinates, etc. Several APIs in authoring environments, plugins and DMC systems offer best practice checking. One of these plugins is the "BIM interoperability Tool" in Revit that offers a list of best practices checks as shown in Figure 10.

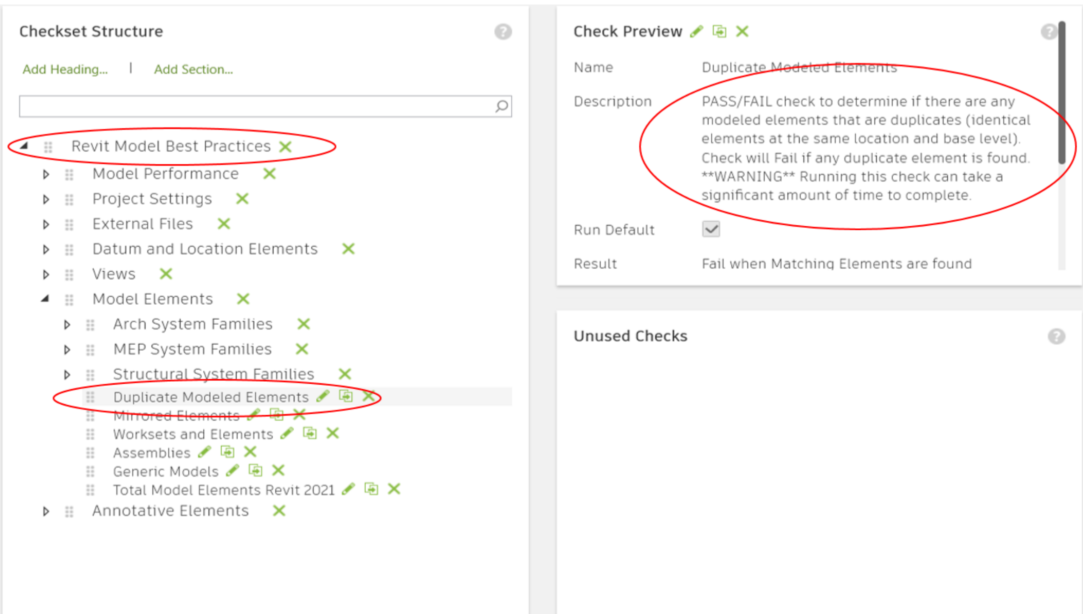

Figure 10 - Example of best practices checks

4. **Exchange Standards** are an assembly of specifications to facilitate interoperability in the AEC industry using collaboration formats. Some definitions of interoperability are as follows:
    - "*ability of two or more systems or components to exchange information and to use the information that has been exchanged*" (ISO25964-2:2013).
    - "*the ability of multiple systems with different hardware and software platforms, data structures, and interfaces to exchange data with minimal loss of content and functionality*" (Guenther and Radebaugh 2004).
    - "*characteristic of a product or system, whose interfaces are completely understood, to work with other products or systems, at present or in the future, in either implementation or access, without any restrictions*". (Wikipedia)

In general, exchange standards are based on two levels of definition. The higher level is concerned about the model schema, and the other is the schema language. Most of the recent exchange formats are based on 4 schema languages (SQL, EXPRESS, STEP, XML). However, brief characteristics of six basic exchange methodology standards are enlisted in Table 4.

Exchange standards are aiding the DMC process by offering Industry Foundation Classes (IFC) as a neutral model representation that is supported by the majority of DMC systems and checking platforms (Eastman *et al.*, 2009). Rule checking using native data models will probably also emerge.

The way exchange standards have become a source of rule-sets is centered on the principles, requirements and specifications of each standard. The rules required to verify and validate COBie (National Institute of Building Sciences, 2007)data files, for instance, are defined in the specification. As per BIMe's (Succar, 2017) Dictionary, the extended definition of COBie states that "*The rules required to verify and validate COBie data files are defined in the specification. These rules ensure that the format of the information provided is consistent, and that the contents mirror existing design and contract documents containing building equipment information.*" This applies a sort of rule that affects DIMs targeting maintainable model elements and its associated semantic and graphical information.

Another application of using exchange standards as a source of rule-sets is the Model view definition (MVD) where generated views rely on the rules that guide to the aimed model view. It normally comprises the kinds of geometry, the critical variables for the use, and the restrictions of object subtypes (Solihin and Eastman, 2015). Such rules and outcome views are usually elevated as per the arrangements among stakeholders and it may also affect the preparation of DIMs prior to the DMC process.

Translation of exchange standard rules requires relatively simple nodes and modules similar to "filter", "selection", "get position", "count number" "Is included", "Is excluded" and "Is Existing". These nodes are user friendly and do not require special domain knowledge and skills. Furthermore, several systems have been performing these simple operations in many forms either web-based as "IFC-checker" (bimspot, 2005), standalone software as "simplebim" (Datacubist, 2009), or plugin as "BIM interoperability tool" into Revit.

Table 4 - Six basic methodology of exchange standards

What it does	Name	Brief	Standard	Schema Language
Describe Processes	Information Delivery Manual (IDM)	-A subset of the overall IFC schema to describe a data exchange for a specific use or workflow. -Looks at individual processes and maps them to understand what should be flowing and when.	ISO29481-1:2016	NA
Information exchange	Industry Foundation Class (IFC)	-Specify a data schema and an exchange file format structure. -The basic 'operating system' that transports the information and data.	ISO16739-s1:2018	EXPRESS XML
Coordination	BIM Collaboration Format (BCF)	-Provides the ability to change management through issue tracking and allows a fully managed cycle.	BuildingSMART	XML
Mapping of terms	buildingSMART Data Dictionary (bsDD)	-Previously known as IFD (international framework for dictionaries) -Collects the terms, vocabulary and attributes.	ISO12006-3:2007	NA
Translates processes into technical requirements	Model View Definition (MVD)	-A subset of the larger model for a particular purpose or use. The items that 'Users' tend to utilize.	BuildingSMART	NA
Information exchange	Construction Operations Building information exchange (COBie)	-Exchange specification for the life-cycle capture and delivery of information needed by facility management (FM)	ISO15686-4:2014	STEP ifcXML SpreadsheetML

4.2.2. Applications

Much of the current applications of rule-sets are desired across industries and information systems. DMC applications dictate flawless identifications of uses, resources and system requirements, performer's competencies and potential output. However, DMC application is a wide expression that has to be built based on a solid basis and covers all possible checking processes

DMC Applications such as code validation, clash detection, health and safety analysis, and interoperability checks seem to be more common and essential. Further elaboration on these applications and previously explained guidelines has derived to the conclusion that DMC applications could be part of the frequently recognized BIM uses. The term "BIM Uses" has been commonly used recently to describe a method of combining explicit BIM benefits that are planned to step into the project life cycle. In this section, instances of DMC applications will be presented and mapped in conjunction with two formalized classifications for BIM uses. These classifications are BIMe Initiative Model Uses List (Succar, 2017), and Pennstate BIM uses (Kreider, R.G., Messner, 2013). These instances have been selected taking into consideration sources of rule presented previously as illustrated in Table 5.

Table 5 - DMC Applications VS BIM uses

Application	Source of Rule	BIMe	Pennstate
Code Checking & Validation	Standards, Codes and guidelines	✓	✓
Risk and Hazard Assessment		✓	X*
Fire and Life safety		✓	X*
Accessibility Analysis		✓	X
Compliance with Project Delivery Documents	Project Documents	X	X
Clash Detection	Best Practice	✓	✓
Model Quality Checks	Best Practice	X	X
Interoperability checks	Exchange standards	X	X

(*) means that the use may be included under code&checking and validation.

The descriptive examples of Table 5, drawn from various rule sources, review a list of the common application. The observations possibly will be outlined as follows:

- Existing BIM uses ontologies that do not fit the DMC applications.
- BIMe list seems more compliant with DMC applications.
- BIMe offers custom model uses on category III that may be expanded to include DMC applications.
- Pennstate compromises BIM uses in line with project phases that may add value to the DMC process by linking DMC applications with the project phase and LODs.
- There is a need for whether including DMC applications within BIM uses and consequently expands existing designations or initiates independent ontology that is dedicated to DMC applications.

DMC application is the first motion that should be outlined, so the other steps would follow. It is also the output that ensures DMC milestones have been accomplished. Furthermore, the source of rules cannot be identified unless the application is delineated taking into account the other impact on the LODs as a certain application may require higher or unified development of the semantics or the graphical information. This is to say that DMC applications are playing a pivotal role in the rule formulation and correspondingly the whole DMC process. Due to the significant role of DMC applications and the lack of formal definitions presented in Table 5, the current representation of DMC applications mandates considerable improvements through terminology that comprise the following recommendations:

- DMC applications may have to be widespread in the means of initiating a separate terminology to contain all possible rule sources.
- This terminology should provide a sort of breakdown to ease the processing of the major DMC applications such as Code Checking & Validation that could contain thousands of applications and rule-sets.
- The terminology has a duty to express the application’s definitions in line with the influencing factors such as project phase, LODs, source of rules and role of the accountable stakeholder.
- The DMC applications are preferable to be connected with checking types in order to express the expected outcome.

4.2.3. Rule Scenario (Clustering)

This section will investigate a set of circumstances that are used to explore Rule-sets and to assist in interpreting the inferences on the DMC process. (Solihin and Eastman, 2015) figured out the criteria of possible rule scenarios that can be categorized into four classifications that illustrate in detail the formulation of rules and associated complexity degree. The upcoming classification is requested to aid the DMC process by emphasizing the requirements of the rule-set that impacts the type of the DMC and the ability to be consistent with the engaged system. These four classifications are:

1. **Class 1 –** rules that need a single or a small number of explicit data.

This means that DMC can be executed directly through specific available attributes or references from the dataset of DIMs. For example, a rule concerns about fire-rated wall nesting doors and windows that should be fire-rated either. To execute this rule:

- DIM should have model elements (window and door) that are connected to the nesting wall and defined fire-rating properties.
- DMC system should be able to perform some types of operations similar to "isConnectedTo", "getFunction" and "EqualTo".

Other examples may come up such as model view definitions. As per (Solihin and Eastman, 2015), rules associated with this class may be originated from building codes, standards and guidelines, exchange standards, or best practices. Throughout the execution of class 1 rules, there is no impact on DIMs as it does not require any further amendments to DIMs. As well, it may involve a single model check as it is regularly intra-disciplinary. However, the outcome of checking the results of class 1 may require slight modifications to DIMs which are mostly semantic more than geometric, and most probably it is just a single loop of review and update without the need for many iterations. Automation of the revising process of this class seems conceivable and safe as it does not require significant involvement of users to implement their knowledge or complicated tracking system for alterations.

Several DMC systems, such as Solibri Model Checker or "BIM interoperability" plugin by Autodesk, are able to support such class of rules due to the limited actions required at model preparation, smooth rule interpretation, and respectively the simple operations required for execution. Also, it is important to mention that this class of rules could exist along various LODs and project phases.

Class Requirements:

This class has a list of requests that have to be guaranteed. These requirements are enlisted as follows:

- Single model check (Intra-disciplinary).
- DIM's attributes and properties should be well-defined (Integrity of Attributes).
- Authoritarian relations among DIM's elements (Integrity of Relations and Constraints).
- Systems should be able to perform simple nodes and modules, and it may allow for automatic updates on DIMs as a privilege.
- Accurate tracking system for changed model elements, if any.

2. Class 2 - rules that require simple derived attribute values

These class includes the rules that are created through stretched characteristics across an advanced level of semantic content, but not generating new data structure. Class 2 rules can be derivate basically through the model element's properties.

For example, ADA compliance checks for a rotating space where a single wheelchair must be able to rotate freely inside a bathroom. For this kind of motion, a clear floor space of at least 60 inches in diameter is required, allowing a 180-degree turn. This rule will be executed by finding the maximum diameter of a circle, not interfering with other geometries, inside an enclosed polygon that represents the boundaries of the ADA accessible toilet. The diameter value will be compared against the 60 inches allowable value given by ADA rule and inferring compliance to two main conditions that are "EqualTo" or "GreaterThan". Besides, this example doesn't need any sort of sophisticated 3D solid modeling for rule execution.

This class of rules is not limited to the given example as it may be also subject to multiple rules that include a number of combinatorial probabilities. Nonetheless, the most common application of this class is multi-disciplinary coordination checks (Clash detection) where several single DIMs are combined and may have the same LOD. Based on the previously given examples, this class correspondingly could be deemed the utmost collective class among the four classes of rule-sets.

Similar to Class 1, this class has no influence on DIMs during the execution phase, and the effect appears only while revising the DIMs according to the outcome of the checking results. In comparison with class 1, the adjustments prompted by Class 2 are reasonably higher, and correspondingly the number of iterations till an appropriate conclusion of the DMC process is more. Nevertheless, allowing DMC systems to perform automatic corrective actions appears dangerous due to the numerous variables in place, by means of the necessity for engaging user's knowledge, and also reasonably the high number of proposed rectifications. Automation controlled by the authorized driver and adequate traceability of changes is probably a conventional solution that avoids data loss and discoordination.

Some of the existing checking platforms can be a satisfactory representative for the execution of this class of rules. On top of these platforms come SMC (Solibri, 2016) and CORENET ePlanCheck system in supporting and checking this class of rules.

Class Requirements:

- Derive new information and calculate multiplex properties.
- Several Single model check (inter-disciplinary).
- DIM's attributes and properties should be well-defined (Integrity of Attributes).
- DIMs at the same level of development (Integrity of Information).
- Consistent relations among DIM's elements ((Integrity of Relations and Constraints).
- Systems should be able to perform composite nodes and modules, with possible minor use of domain Specific areas.
- Integration between a solid tracking system and coordination among stakeholders to ensure collaborative interaction (Data Collaboration).

3. Class 3 - rules that require an extended data structure.

(Solihin and Eastman, 2015) extended the classification of rules by articulating the natural development of previous classes 1 and 2 that has led to a class that requires a high level of geometric and semantic conditions, and typically extensive computations. Therefore, there is a need to extend the structure of data sets in DIMs to perform the DMC process.

The three-dimensional relationship of an overall building can be an elaborative example of class 3, the relations between certain objects and attributes are implicit and necessitate further operations to establish complex topological structures for the interpretation process. To illustrate, the smoke detectors spacing rules of NFPA 72 (National Fire Protection Agency, 2013), Chapter 17, specifically Section 17.6.3.1.1, where detector layout must follow one of the two requirements below:

- *The distance between detectors shall not exceed their listed spacing, and there shall be detectors within a distance of one-half the listed spacing, measured at right angles from all walls or partitions extending upward to within the top 15 percent of the ceiling height.*
- *All points on the ceiling shall have a detector within a distance equal to or less than 0.7 times the listed spacing (0.7S).*

The system concerned for checking this rule has to detect the positions of smoke detectors, which has a listed spacing of 30 feet, within the 3D space, and to ensure compliance with the design in Figure 10 that match with both requirements. The detector is no more than one-half it is listed spacing from any wall measured at a right angle. In addition, no space on the ceiling is more than 0.7 x the detector's listed spacing, in this case, 21 feet. An analysis diagram using a distance triangulation is mandatory to achieve the requirements of this rule that allows detecting the other nearby detectors and therefore the distance between detectors can be checked.

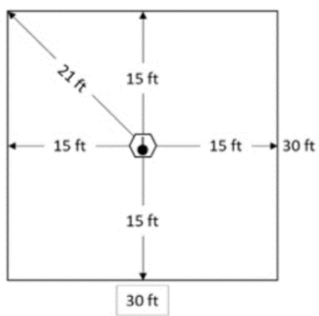

Figure 11 - Illustration of the spacing rule of NFPA 72

The most important source for class 3 rules is standards, codes and guidelines. As well, there is an influence on DIM's data structure in terms of calling for higher LOD and the prerequisites of complex computations. Therefore, DIM authors should consider the possibility of supplementary semantic and graphical information and system developers should allow for more computations in the authoring environments which clashes with proprietary files that usually are protected against external encoding. This is another factor that causes increased acceptance of Industry Foundation Classes (IFC) files in the DMC process, in addition to the efforts towards open BIM.

Given examples of Class 3 may endorse systems such as FORNAX, as implemented in CORENET ePlanCheck, and GSA building design circulation checking based on SMC which has requisite basics for the execution of class 3 rules.

Class Requirements:

- Extended Data Structure.
- May derive a new model to enable the valuation of certain implicit attributes or relations.
- Single or Several Single model check.
- DIM's attributes and properties should be well-defined (Integrity of Attributes).
- Moderately high level of development (Integrity of Information).
- Due to the looping effect, the exchange of DIMs thru open standard is desirable (e.g. IFC).
- Systems should be able to perform multiple complicated nodes and modules.
- Existing checking platforms may be not able to satisfy the user needs, thus major use of domain Specific areas may be present.
- Solid domain knowledge.
- Integration between a solid tracking system and coordination among stakeholders to ensure collaborative interaction (Data Collaboration).

4. **Class 4** - rules that require a "proof of solution".

The rules of this class are used for reviewing a design instead of detecting compliance or non-compliance checks. In other words, class 4 is more concerned with solutions having more than one acceptable answer. (Solihin and Eastman, 2015) added that although DIMs are subject to additional information to propose a solution, the proposed design options are usually either temporary or virtual till the concerned driver decides either one of the proposed designs is approved or rejected. Furthermore, the type of applications using rules of class 4 is merely performance-based. Because of that, it is worth mentioning that the source of class 4 rules may include the massive "knowledge database" mind from best practices over and above codes, standards, and guidelines. This sort of solution is at a much-undeveloped level within the AEC industry, and dedicated systems resolutions are not known (Hjelseth, 2016).

Screening probable alterations of class 3 may develop into class 4. To illustrate, the influence on DIMs and systems considering the complexity of class 4 rules is almost similar to the cause-effect of the other classes; however, seizing knowledge and present it may add few requirements. Rules of class 4 may interrupt both design and construction phases and also throughout several LODs which means systems should be well-prepared to deal with a relatively high amount of data and also more complex computations.

Class Requirements:

Refer to the requirements of Class 3.

4.2.4. Checking Types

This section is concerned about defining the types of DMC that safely drive rule-set requirements to a system for execution. (Hjelseth, 2016) published a study that offers an ontological method to classify unique types of DMC. This research will try to present these types in conjunction with the rule scenarios exemplified previously. This dual interpretation aims to end up with a higher understanding of what are the required competencies of the DMC system. The method has identified types of DMC as follows:

1. Compliance Checking Solution, divided into

a) Validation Checking

The definition given by (Hjelseth, 2016) is to verify that DIM designated content is in line with the constraints of rule from a rule-set. However, rule classes 1 and 2 give the impression of being the best possible participation for validation checking. The logic of checking here is illustrated in Figure 12 where the constraints in the rule-set must be larger than DIM content (indicated by the area of the circles) and the result can be "Pass", "Fail" or "Not Checked" providing that the logic has not requested any extended data structures on the DIM content side, and irrespective of any complicated computations. Taking into account that the only conservation interrelated to DIMs is the necessity to have the same LOD.

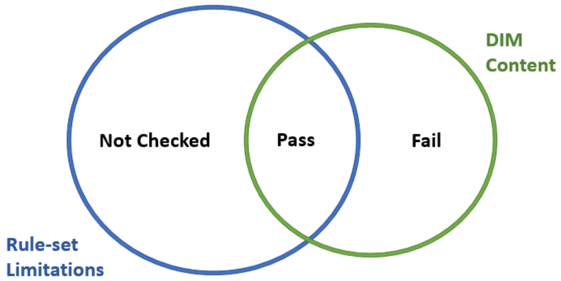

Figure 12 - Logic of validation checking

(Hjelseth, 2016) added that validation checking is either geometrical based (e.g. Class detection) or Information based (e.g. Code Compliance). However, given examples on each type are probably deficient since code compliance checks may also require to be geometrically based.

Requirements of a system concerned with validation check are merely bounded with rule requirements that do not affect the data structure of DIMs and should give out "Pass", "Fail" or "Not Checked" as being the checking results. However, validation checks may be regarded as a DMC type that directives several feedbacks looping mechanisms.

Furthermore, validation checking may take place using an internal feature within the authoring environment similar but not limited to "interference check" in Revit, domain-specific areas, or independently at standalone proprietary checking systems such as SMC. Updates by given approaches are not permitted to overwrite the original DIM unless further authorization by the concerned stakeholder.

b) Content Checking

Where the content of DIMs has to be tested against a particular application or a list of terms by just applying "filters" that delineate pertinent information. Bearing in mind the logic of content checking as shown in Figure 13 is providing that the circle indicating the DIM content has changed after conducting the checking process by the deletion of element “B” which means DIM’s data-in are typically larger than or at least equal to DIM’s data-out. In other words, the input model has a practical status of “in-progress” while the output is “shared”.

In Figure 13, the list possible of actions could be summarized to 1) checking result which is either “identified” or “not identified”, 2) compliance report (e.g COBie spreadsheet) or 3) omission of unlisted irrelative information. These characteristics are matching with class 1 of rule scenarios. This announced logic has not changed the structure of information or even requested any sort of supplementary computations. However, the LOD of DIMs on this type does not affect the process as it only touches the quantity of identified or not identified information with respect to the enlisted rules and relations.

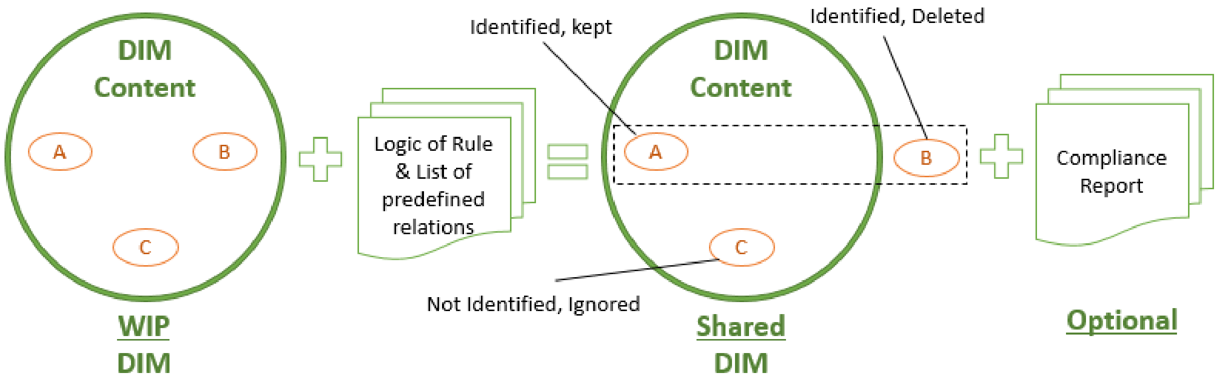

Figure 13 - Logic of content checking

The famous examples of content checking as per (Hjelseth, 2016) include applications and rule-sets obtained from exchange standards that are named “Model completion checking” and another filtering check that named “Protection checking”.

Content checking does not necessitate high domain knowledge or the extreme contribution of software developers due to the explicit operations and computations associated with the execution of rule-sets of this kind of checking. A system such as “Simplebim” or is capable of performing this sort of checking smoothly and efficiently.

The main feature here is that the newly created DIM has to not overwrite the original file since this means permanently losing data from the original DIM that can not be easily retrieved. Therefore, DMC systems must not allow for overwrites under any circumstances. In most of the case, content checking may also not require more than a single loop of feedback mechanism or most likely to be a one-time check.

2. Design solution checking, divided into:

c) Smart Objects Checking:

(Hjelseth, 2016) outlined the objective of smart object checking that is to implement adaptive objects in DIMs. These objects are pursuing to automatically adjust itself and surrounding model elements exploiting entrenched pre-defined rules and algorithms. These adaptive objects have their own semantic and graphical information that is also required to blend with existing Graphical and non-graphical information of the DIMs. The illustration is shown in Figure 14 emphasizes on the previously given definition of smart objects checking where the input is 1) DIM, 2) adoptive object and 3) rules or/and algorithms and the output is 1) extended DIM content either semantically or graphically and 2) Adaptive object that has new characteristics (presented by the size of the rectangle).

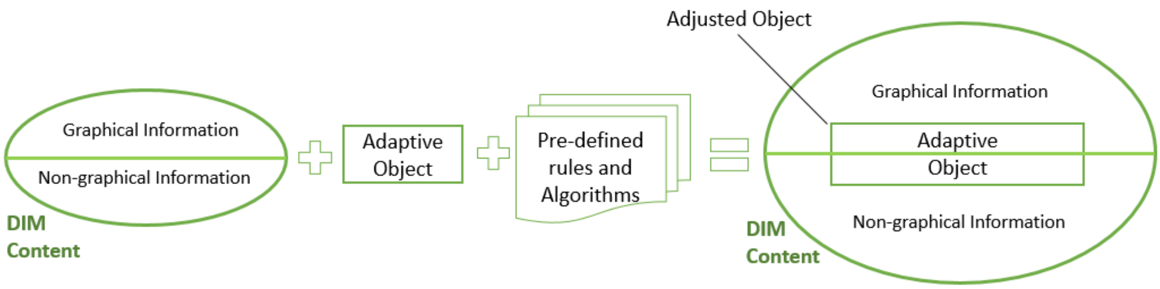

Figure 14 - Logic of smart objects checking

The straightforward example of this type of checking that illustrates the logic could be when creating a Revit family using an adaptive family template. The input on this example is an adaptive Revit Family such as a window family that usually behave according to the logic specified by the smart object checking, a hosting wall in the DIM, and an optional rule given by Revit and named “formulas”. If the window family is not hosted by a wall or the dimensions of the window exceeds the dimensions of the wall, Revit will show an error message that has to be rectified immediately. The earlier example only exemplifies the logic of smart objects checking that considers only the graphical information of the adaptive objects; however, it is worth mentioning that this checking is not limited to graphical information but also semantics such as the specification of material properties or fire-rating properties.

Theoretically, the use of smart object checking may involve any kind of checks that has the interest to examine the DIMs behavior against extended data structures, therefore, the optimal class of rules associated with this kind of check is Class 3. However, the practical applications of smart object checking may encompass constructability analysis, and the risk assessment of health, safety, and environment (HSE). The rules of these applications are commonly sourced from best practices, standards, codes and guidelines.

Systems involved with smart object checking should allow for immediate implementation of the checking output either by overwriting the original DIM or creating new DIMs due to the relatively high number of iterations, feedback mechanism and justifications required prior to the official implementation that involves also user’s knowledge.

d) Design Option Checking:

Apply external knowledge in a form of building solutions to the DIMs and then comparing the result to find the best optimal solution is the definition of design option checking. The guidance given by design options is relevant to rule-sets that identify predefined schemes. The illustrated logic of design options checking is shown in Figure 15. To exemplify, the existing solutions originally in the DIM are "B" and "C" while the proposed solutions are "E" and "D". The relation between DIMs and external knowledge is two ways. The first way is when the DIMs obligates some rules on external knowledge, and the other way is when external knowledge supplies DIMs with proposals.

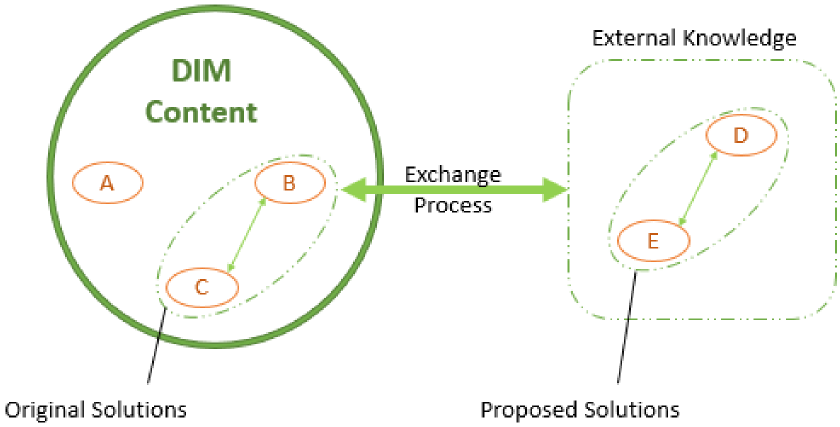

Figure 15 - Logic of design option checking

External knowledge is relatively a huge data-base that could be gathered form best practices, standards, codes and guidelines. As design option checking is not about compliance or non-compliance, the perfect match for design option checking is Rule Class 4 that shares the same objectives. On the other hand, since exchanged proposals are hard to be predictable, systems concerned with design option checking are still limited and require further developments.

(Hjelseth, 2016) pointed out a viable question about the capacity of the proposed solutions. The question is "if all other options are checked out", and the answer is "it depends" on the solidity of the proposed solutions, but in all cases, partial support is acquired providing that the feedback loop mechanism is already included in the ongoing exchange process.

4.3. DMC Implementation Plan and Associated Risks

The unique contribution of this research is the method followed along with the implementation plan. This section will present steps of the DMC implementation plan followed by the associated risks as follows:

4.3.1. DMC Implementation Steps

This section will assemble predefined extended concepts of DMC principles into a workable plan.The implication of each concept will be underlined over the five steps. The series of steps shown in Figure 16 will consist of: DIM Prerequisites, DIM preparation, Rule interpretation, DMC Execution and Checking Results.

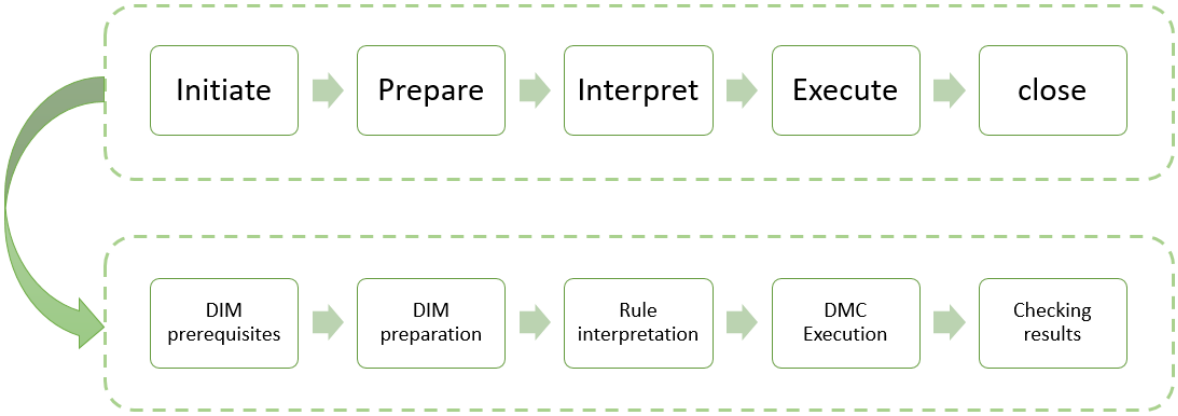

Figure 16 - DMC Implementation Plan

DIM Prerequisites

This step considers as the kick-off point of the quality management of DIMs that evolve proper documentation of the DMC implementation plan. Documenting the DIM prerequisites step is as important as the Post-contract BIM Execution plan (BEP) since both acts as project rules that preserve stakeholders updated and involved concerning the accomplishments and quality of the running project.

The preconditions of the DMC process require full readiness of the DIMs that could be achieved by proper enactment of rule requirements and system requirements. Therefore, this research proposes that the documentation of DIM prerequisites should express 1) the objectives of the DMC and 2) what is needed to achieve DIM readiness. The illustration of the documentation process and involved concepts is shown in Figure 17.

The DIM prerequisites start with defining the objective of the DMC process in order to define the source of rules and consequently categorize the rule-sets. According to the class of rule-set, the rule requirements and checking type will be clearly outlined announcing a list of necessities for the rule-set. The system requirements are not limited to checking types but also it may include interoperability shortcomings among involved systems that have to be outlined early in the DMC process.

On the other side, the handover acts as a contract between the DIM author and DMC driver that should undertake data collaboration, integrity and possible developments of the LOD according to the class of rule-set. As soon as requirements and hand-over processes are outlined, the outcome of step 1 is a list of actions either to the DIM author or DMC performer based on the predefined responsibility matrix. This means the completion of step 1 and moving forward to step 2 which is “DIM preparation”.

The aim of step 1 is to pave the road of DMC implementation thru sorting out possible complications and provide proper documentation of these obstacles that are subject to further developments throughout the life cycle of the DMC process. Formal documentation of this step is mandatory and may be attained through attaching a new section to the project delivery documents or as a new standalone document.

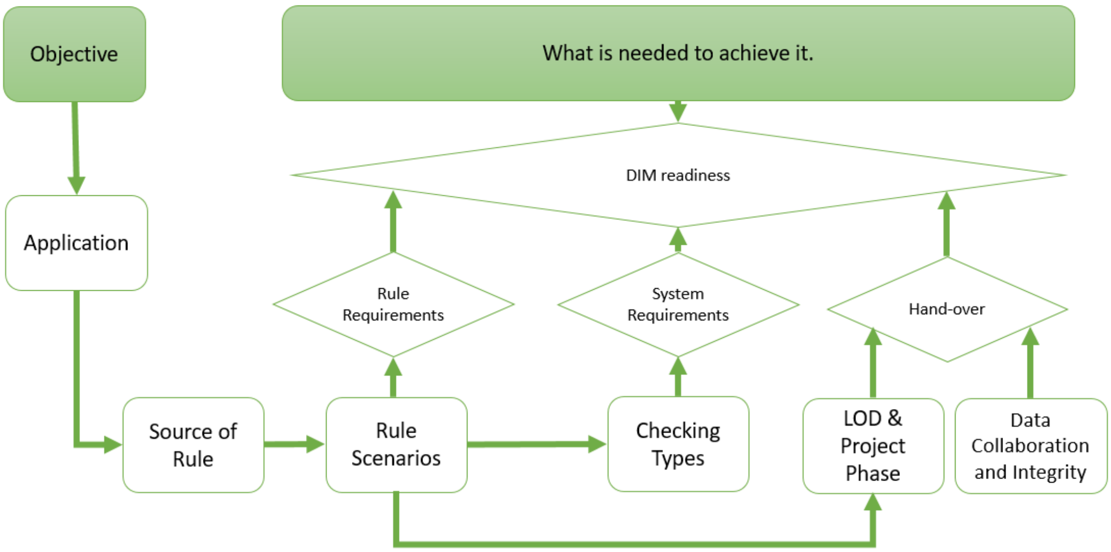

Figure 17 - DIM Prerequisites

DIM preparation

Step 2 reveals the tangible arrangements to achieve DIM readiness. The list of actions outcome from Step 1, DIM prerequisites, are the input for the DIM preparation step.

As per (Eastman *et al.*, 2009), the five possible activities for DIM preparation are outlined with definitions in Figure 18 and as follows:

1) *“Model Views”.*
2) *“Derive implicit properties using enhanced objects”.*
3) *“Derive new models”.*
4) *“Performance-based model and integrated analysis”.*
5) *“Visibility of layout rule parameters*”.

Further reading into these preparation activities, may tell that it’s merely could be mapped against rule requirements (sourced from Rule scenarios) that has been documented at step 01. The illustration of these mapping processes is shown in Table 6.

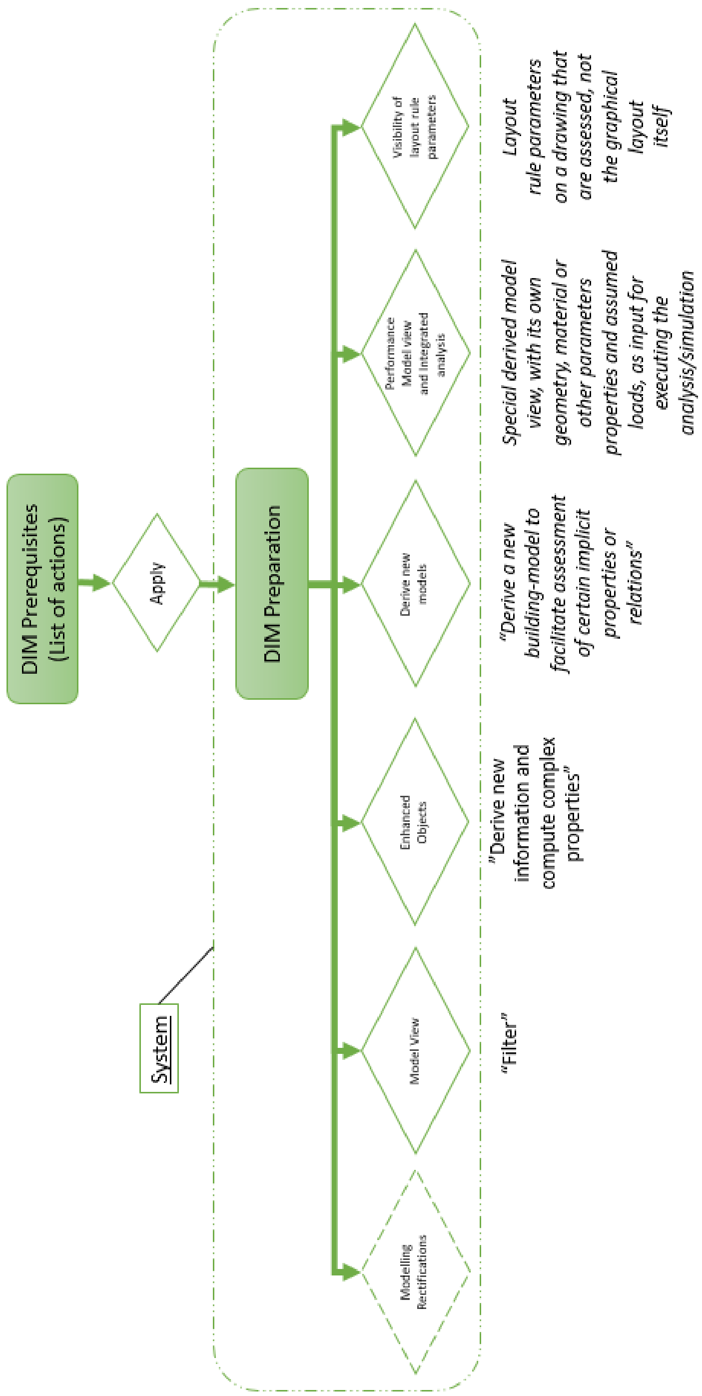

Figure 18 - DIM preparation

Table 6 - Mapping DIM Preparation VS Rule Scenarios

DIM Preparation	**Rule Scenario**
Model View	Class 1
Enhanced Object	Class 2
Derive New Models	Class 3
Performance-based model and integrated analysis	Class 4
Visibility of layout rule parameters	**Not Mapped**

As distinguished in Table 6, the "Visibility of Layout rule parameter" is not mapped. However and as described by his activity is more related to rule restrictions on layouts to check the parameters in the drawing than every instance of a model element's properties, accordingly generate modernized layouts controlled by rules. In this context, the mismatch comes along the fact that that IFC schema languages, XML and EXPRESS, cannot denote layout generation rules. This activity has tried to manipulate the preparation step in 2D rather than 3D which seems acceptable with limitations.

DIM preparation activates should occur within a system that may or may not be the same system for the DMC execution step. However, this will raise the concern of how the prepared DIMs should be stored for further use and the answer is whether to generate separate physical DIMs or virtually extend the data structure on the original DIMs. Either solution should not allow for overwriting on the original DIMs unless it is authorized by the concerned stakeholder.

Major observations may discredit the preparation activity. The first is related to modeling methodology and ensuring proper data integrity among various disciplines and stakeholders. This activity has been documented at step 1 under the Hand-over sub-process. That's why this research may add a new layer to the preparation list which is "Modelling Rectifications". The second is interrelated to the system requirements where it has no major impact on the DIM preparation step unless the interoperability among involved systems is not in place providing that the IFC schema may overpower such constraint.

Rule interpretation

Rules in an any sort of documentations are in an original textual form that is not suitable for implementation in a DMC system. It is required to be formalized as a machine-readable rule to be presented in the form of logical expressions. This layer is deployed to perform this sort of transformation noting that major operations are domain-specific which require solid domain knowledge. This simplified approach is presented in Figure 19.

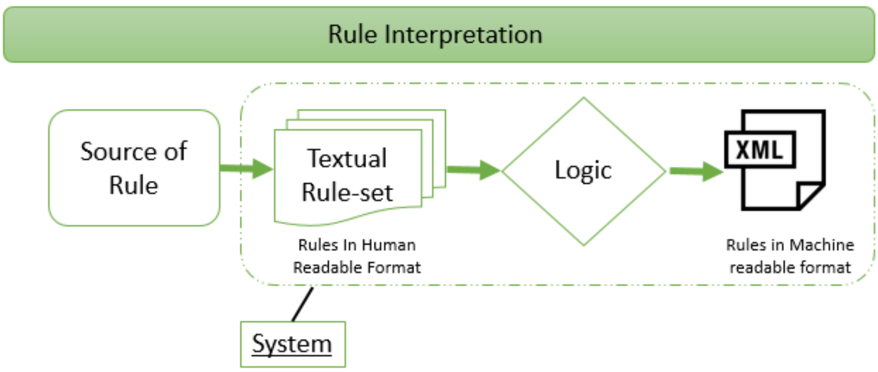

Figure 19 - Rule interpretation

DMC Execution

This step is an intermediate executive station between exploiting the inputs and generate a viable output. The input of this step has been gathered from step 2 (DIM preparation) and step 3 (Rule interpretation) in order to be operated in a DMC system. Although preparation and interpretation steps in line with the predefined DIM prerequisites are the engine and fuel of the execution step, checking types are also another regulator that leads to properly choose the adequate system for DMC execution. These interrelations are shown in Figure 20. The main challenge in this step is to match checking types with an effective system. Table 7 offers guidance of what is the system type that may potentially aid a certain checking type.

Table 7 - Guidance for choosing a system based on Checking Type

Checking Type	System
Validation Checking	Proprietary framework Hard-Coding Domain-Specific Area
Content Checking	Proprietary framework Hard-Coding
Smart Objects Checking	Proprietary framework Domain-Specific Area
Design options Checking	Domain-Specific Area

In summary, well-prepared and interoperable data-sets, the executable format of interpreted rule, targeted checking type and system able to participate in all previous conditions are all what a stakeholder needs for DMC execution. Completion of the DMC execution step is announced when checking results are present, and this means the kick-off step 5.

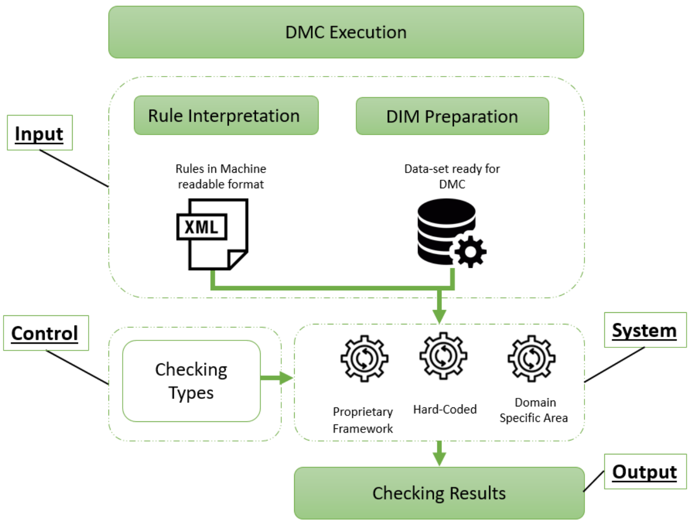

Figure 20 - DMC Execution

Checking results

The final station of the DMC process that declares states of the resultants. The output of this layer may either announce officially close-out of the DMC process or mandate updates on the DIMs. If it is a single loop iteration similar to the results from Model content checking, it means no further updates are required on the DIMs. The other checking types are subject to several loops of updates that happen in accordance with the data feedback mechanism until effectively close-out the DMC process. These denotations are illustrated in Figure 21 where also all possible checking results are presented with respect to the checking type.

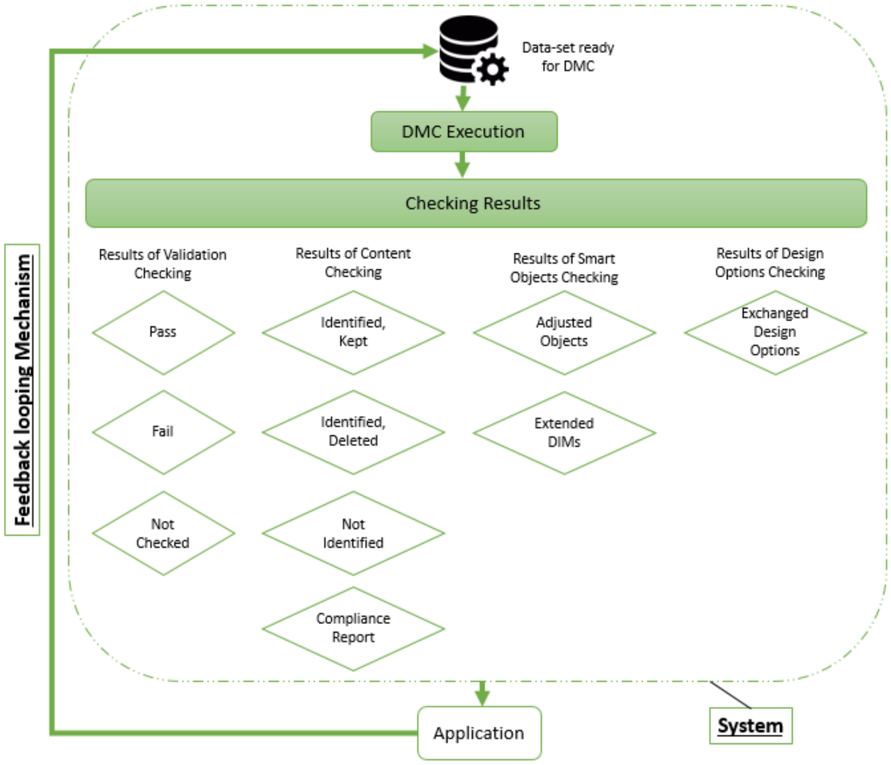

Figure 21 - Checking Results

4.3.2. DMC Implementation Risks

This section focuses on the risks associated with the DMC implementation plan. These risks are the potential for an improvement or deployment failure. With that in mind, it will be useful to consider the top five risks that the practice may encounter throughout a DMC implementation plane.

Usability

One of the considerable challenges is the usability of the DMC implementation plan between stakeholders. Successful DMC means that the stakeholders are required to initiate the implementation plan, progress accordingly, frequently reflect and update changes, coordinate between each other and finally close-out. The usability of the DMC thru a project life cycle is threatened by the need for continuous attention to the implementation plan throughout the project life cycle. The author of this work argues that large scale or fast-track projects are the more affected types of projects because the involved stakeholders consider other milestones on top of their priorities neglecting the fact that DMC ensures the quality, accuracy and consistency of their end products.

Openness

DMC is a process that rather requires two or more systems to interconnect. Not only systems but also stakeholders are requested to exchange DIMs. Openness is highly mandated between DIMs, systems and stakeholders. Therefore, systems and DIM's native formats should be open source or interoperable to ensure the openness, and consequently a successful data exchange. On the other side, stakeholders should be able to coordinate and work together whether with or without a common data environment. The current situation shows that continuous efforts towards the openness of DIMs, systems, and stakeholders are promising (Zhang *et al.*, 2017). The presence of the IFC schema with its supplementary languages XML and EXPRESS has encouraged several system developers and stakeholders to consider a proper IFC adoption taking into account that it's the optimal neutral digital model representation (Zhang *et al.*, 2017).

DMC implementation plan has presented several exchange activities during the hand-over, interpretation of rules, and execution phase. These activities mainly are dependent on the level of openness of DIMs, systems, and stakeholders. In other words, if the systems and DIMs are not open, this means data deficiencies that threaten the sub-activities and accordingly the whole implementation plan.

Documentation

So far, DMC has not been officially recognized by any standards or regulations as a standalone documented process, or in the same context that outlines the project milestones and deliverables. It may be included in the BEP document as a model use, but then again this underestimates the significance of the DMC process. Moreover, some of the existing ontologies to categorize DMC concepts of this work do not seem ready to be used similar to the inability of BIMe or Pennstate to be mapped with DMC applications. DMC implementation plan as presented in this research, require newly developed Documentation that may include DMC applications and DIM prerequisites from the implementation plan, in addition to, source of rules, rule scenarios, and checking types from the DMC modules. These developed concepts and stages are vital to be documented and standardized which may seem against the willpower of the industry. The author of this work would further elaborate that the industry has a sensitive response towards any kind of change that may be called "Change Resistance", therefore it is also possible to say that the proposed concepts and plan may confront the same resistance.

Complexity of Rules and DMC systems

The development of DIMs to conclude the continuous progress of building systems and to cope against the expansion of digitalization has led to more complex rules. Some DMC applications or sources of rules are already producing complex rules such as Code checking & validation or Health and safety assessment. Complex rules are usually associated with Classes 3 and 4 of rule scenarios where extended data structure and/or complex computation are required. This complexity affects DMC systems and correspondingly two crucial steps of the implementation plan that are rule interpretation and DMC execution. This is to say that the complexity of rules should determine the selected DMC system while in practice the other way occurs. To illustrate, the DMC rules are practically more dominated by checking platforms and plugins capabilities whereas extra requirements, that not within these capabilities, may be checked manually or by relying on the knowledge of the specialists.

Because DMC systems are a wider approach that includes checking platforms and plugins as subdivisions, the implementation plan has suggested using DMC systems instead of checking platforms or plugins. However, DMC systems may recall for domain knowledge that possibly will be able to deal with complex rules. The knowledge and skill level of domains must be taken into consideration in a way that it could be more expensive than the initial budget anticipated.

Stakeholders and Contractual issues

The DMC process may engage several stakeholders who accomplish their duties in parallel or series. This is to say that without a clear responsibility matrix and proper hand-over, the activities that involve several stakeholders may get complicated. This causes contractual implications that once these influences have arisen, it will have the domino-effect, and reach most of the other activities.

5. DISCUSSION

A large number of current approaches and arguments have shown the extraordinary significance of Model Checking for the AEC industry. This importance has led to a growing interest among scholars, researchers and practitioners within an organizational, governmental and public institutes. The review of this work has revealed that DMC is still subject to further developments due to relatively numerous constraints and unexploited aspects. Therefore, this research focused on the DMC, to maximize the potential values and allow for a wide range of applicability among scholars and practitioners.

To address such barriers, the work of this paper had planned for three research questions and then predicted that the answers to these questions will provide the way forward solutions awaited to overcome the DMC imperfections. This section presents a critical analysis of the research findings and follows the structure of answering the three research questions.

The **first** research question aims to survey the current knowledge of DMC and its influence on the AEC industry where the search engine approaches studies that are about e.g. “BIM-based model checking”, “Rule-based checking”, or “Automated Code Compliance Checking Based on a Visual Language and Building Information Modeling”. However, the answers to the first question could be outlined as follows:

1. Several efforts on parts of the checking process that concerns about only converting textual rules into a machine-readable format. (Preidel and Borrmann, 2015) (Zhang and El-Gohary, 2012) (Khan *et al.*, 2020)
2. Place extensive focus on “Code Validation & Compliance” considering it represents model checking. (Nawari, 2012)
3. Defining the required objects, attributes, and relations within an information model that could participate with rule requirements derived from a rule classification. (W. Solihin and Eastman, 2015)
4. Defining the types of model checking and the nature of the expected results without proper linking with other checking parameters that could verify or contradict the feasibility of these types. (Hjelseth, 2016)
5. Limited approaches that supplement the relation between model-checking and project phases. (Galkina and Kuzina, 2018)
6. Frameworks illustrating independent DMC processes that may invite scholars while practitioners will question how to be applied across their other existent processes. (Eastman *et al.*, 2009)
7. Lack of standards and documentation that concerns DMC requirements, sources of rules, process milestones, and possible circumstances (Hjelseth, 2016).

The common between the above approaches that it simply based on three basic principles that are needed for a checking process. These principles are Rule-set, Model, and checking Platform that assume if these principles are achieved, the user will be able to conduct model checking.

The **second** question relates to the developments required to reinforce digital model checking where it becomes vital to expand current concepts and find proper relations that will increase the ability to manage DMC principles (digital information model, rule-sets, and checking platforms) in an integrated and collaborative manner. The answer to the second question mandates a framework that takes into consideration the following:

1. Model-checking is not only about Building information management (BIM) but also about the totality of information that should include existing physical systems not only buildings.
2. Certainty about the applications of model checking to conclude possible uses, not only code validation, in the presence of formal standards.
3. Determining the role of stakeholders along with the checking process.
4. Study the impact of a project phase on model checking.
5. Finding the optimal representation of digital information models.
6. Rules are not only sourced from regulatory codes and standards, and there are other sources of rules that should be taken into consideration.
7. Classification of rules that defines its requirements should be mapped with possible checking types and expected results.

This list discusses primarily the limitations that this research has recognized. However, these limitations are dependent on a large number of circumstances that may not be included in this thesis

Based on this list, the expansion has been achieved through a modularization approach that proposed a conceptual DMC framework that can be structured into three modules named as follows: 1) Digital Model Representation, 2) Rule Formulation, and 3) DMC implementation plan

The **third** question is about the documentation required for a DMC process in the context that considers model-checking as a dependent activity that has cause and effect on former and subsequent activities. This has resulted in an implementation plan where the predefined extended concepts of DMC principles have been assembled into workable steps.

6. CONCLUSION

DMC is a process that still an area of further improvement and not yet completely formalized. This means that there is still a need for enhanced system competences, updated processes, and amendments to the current project delivery documents, and proper standardization of DMC characteristics. To address these obstacles, the work of this paper had intended for three research questions and then predicted that the answers to these questions will provide the major findings and the way forward solutions awaited to overcome the DMC imperfections. The **first** research question aims to survey the current knowledge of DMC and its influence on the AEC industry. The **second** question relates to the developments required to reinforce digital model checking where it becomes vital to expand current concepts and find proper relations. The **third** question is about the documentation required for a DMC process. Therefore, this work proposes a diagnostic approach based on using pre-defined principles to analyze digital model checking (DMC) and propose a formal framework and implementation plan.

The major findings of this research have commenced at an early stage when this study has been utilizing the term "digital model checking (DMC)" instead of "BIM-based checking" or "Rule-based checking". Afterward, a modularization approach has been established to expand the current basic principles of the checking process (Rule-set, Model, and checking Platform). Throughout the expansion of principles, modules were built on a basis that it has not ignored the involvement of stakeholders and the holding machinery system. Furthermore, it has considered the effectiveness of digital information models to integrate, exchange, identify, and verify its content taking into account present classification that could aid the DMC process. However, the expansion has been achieved through three modules that concern about digital model representation, rule formulation, and model-checking implementation plan.

The first module (digital model representation) disassembles the requirements of the digital information models to comprehend important layers that aid the readiness of the information models and consequently the digital model checking. These layers are Data integration, collaboration, Level of development, and project phase. The support given by these layers is to ensure meaningful information that is matured and could be exchanged among stakeholders throughout a project life-cycle.

The second module (rule formulation) analyses the rule formulation thru four layers that outlines the rule applications, sources, classification, and comparable checking types. The objective of this module is to allow users to have full control over the rule-sets by being able to distinguish how rules are sourced, what are the possible uses of the rules, what are the rule classes and requirements in the world of information models and encoding systems, and how to gather all these pieces of evidence to decide on the proper checking type.

The third module (digital model-checking implementation plan) is where the knowledge of modules 1 and 2 have been gathered to produce an effective implementation plan. The plan has five major steps that start with primarily outlining the objectives of the checking process, rule formulation, and model requirements in conjunction with a hand-over activity between the performers of the model checking and the authors of the information models. This first step aims to document and examine the readiness of the information models at the early stages of the model checking. Based on the list of actions of step 1, models will be tangibly prepared at step 2. Interpretation of rules starts with step 3 where the checking rules are extracted from its formal sources and transformed into a machine-readable language. Afterward, the closure of the checking process is on steps 4 and 5 when the execution of digital model checking and checking results have been completed.

With these findings, this study has revealed the benefits of the model checking modularized approach. Digital model checking is still an area of development by the means of enhanced system capabilities, updated processes, adjustments to the current project delivery documents, and proper standardization of concepts. However, there are still open questions about the ability of the industry and international organizations to conclude these developments. The author hopes that further progress in the scope of digital model checking can become more concentrated. The benefits of the modularization approach may be invested in a case study that involves the stakeholders and software developers in a practical approach. Furthermore, formal guidelines thru standards will be an added value to the model checking processes. This aligns with properly considering digital model checking as a process that has its milestones, prerequisites and circumstances. Finally, exchanging lessons learned through feedback loop mechanisms will avoid undertaking the same inaccuracies.

REFERENCES

Autodesk (2020a) Navisworks. Available at: https://www.autodesk.co.uk/products/navisworks/overview?plc=NAVSIM&term=1-YEAR&support=ADVANCED&quantity=1.

Autodesk (2020b) Revit. Available at: https://www.autodesk.com/products/revit/overview.

BCA Singapore (2013) Singapore BIM guide V2. Singapore: Building and Construction Authority. Available at: https://www.corenet.gov.sg/general/bim-guides/singapore-bimguide-%0Aversion-20.aspx. %0A.

BIMForum (2019) 'Level of Development (LOD) Specification Part 1 & Commentary For Buidling Information Models and Data'. Available at: https://bimforum.org/wp-content/uploads/2019/04/LOD-Spec-2019-Part-I-and-Guide-2019-04-29.pdf.

bimspot (2005) ifc-checker. Available at: https://ifc-checker.com/.

Borrmann, A. et al. (2018) Building information modeling: Why? What? How?, Building Information Modeling: Technology Foundations and Industry Practice. doi: 10.1007/978-3-319-92862-3_1.

BuildingSMART (2009) BIM Collaboration Format (BCF). Available at: https://technical.buildingsmart.org/standards/bcf/.

BuildingSMART (2020) IFC Schema. Available at: https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/.

database.guide (2016) What is Data Integrity? Available at: https://database.guide/.

Datacubist (2009) simplebim. Available at: https://simplebim.com/.

Dimyadi, J. and Amor, R. (2013) Automated Building Code Compliance Checking - Where is it at? doi: 10.13140/2.1.4920.4161.

Eastman, C. et al. (2009) 'Automatic rule-based checking of building designs', Automation in Construction. Elsevier B.V., 18(8), pp. 1011–1033. doi: 10.1016/j.autcon.2009.07.002.

Fan, S. L., Chi, H. L. and Pan, P. Q. (2019) 'Rule checking Interface development between building information model and end user', Automation in Construction, 105(October 2018). doi: 10.1016/j.autcon.2019.102842.

Galkina, E. and Kuzina, O. (2018) 'Building information model verification at the lifecycle stage of construction', IOP Conference Series: Materials Science and Engineering, 365(6). doi: 10.1088/1757-899X/365/6/062031.

Getuli, V. et al. (2017) 'BIM-based Code Checking for Construction Health and Safety', Procedia Engineering, 196(June), pp. 454–461. doi: 10.1016/j.proeng.2017.07.224.

Ghannad, P. et al. (2019) 'Automated BIM data validation integrating open-standard schema with visual programming language', Advanced Engineering Informatics, 40(March), pp. 14–28. doi: 10.1016/j.aei.2019.01.006.

Hjelseth, E. (2015) ‘BIM-based model checking (BMC)’, in Building Information Modeling: Applications and Practices, pp. 33–61. doi: 10.1061/9780784413982.ch02.

Hjelseth, E. (2016) ‘Classification of BIM-based Model checking concepts’, Journal of Information Technology in Construction (ITcon), 21, pp. 354–369.

Hjelseth, E. and Nisbet, N. (2010) ‘Overview of concepts for model checking’, Proceedings of the CIB W78 2010: 27th International Conference –Cairo, Egypt, (September), pp. 16–18.

ISO12006-3 (2007) Building construction — Organization of information about construction works — Part 3: Framework for object-oriented information. Available at: https://www.iso.org/standard/38706.html.

ISO15686-4 (2014) ‘Building Construction — Service Life Planning — Part 4: Service Life Planning using Building Information Modelling’. Available at: https://www.iso.org/standard/59150.html.

ISO16739-1 (2018) Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries — Part 1: Data schema. Available at: https://www.iso.org/standard/70303.html.

ISO19650-1 (2018) Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 1: Concepts and principles. Available at: https://www.iso.org/obp/ui/#iso:std:iso:19650:-1:ed-1:v1:en.

ISO25964-2 (2013) Information and documentation — Thesauri and interoperability with other vocabularies — Part 2: Interoperability with other vocabulariesNo Title. Available at: https://www.iso.org/standard/53658.html.

ISO29481-1 (2016) Building information models -- Information delivery manual -- Part 1: Methodology and format. Available at: http://www.iso.org/iso/catalogue_detail.htm?csnumber=60553.

Khan, N. et al. (2020) ‘Visual language-aided construction fire safety planning approach in building information modeling’, Applied Sciences (Switzerland), 10(5). doi: 10.3390/app10051704.

Khemlani, L. (2005) CORENET e-PlanCheck: Singapore’s automated code checking system. AECbytes. Available at: http://www.novacitynets.com/pdf/aecbytes_20052610.pdf.

Kreider, R.G., Messner, J. I. (2013) The uses of BIM. Classifying and Selecting BIM. Available at: https://www.bim.psu.edu/bim_uses/.

Lee, J. M. (2010) ‘Automated Checking of Building Requirements on Circulation Over a Range of Design Phases’, Building, p. 172. Available at: http://smartech.gatech.edu/bitstream/handle/1853/34802/lee_jae_m_201008_phd.pdf;jsessionid=10FA65D2354A62739D238AB3B3A15577.smart2?sequence=1.

Mekawy, M. and Petzold, F. (2018) ‘BIM-based model checking in the early design phases of precast concrete structures’, CAADRIA 2018 - 23rd International Conference on Computer-Aided Architectural Design Research in Asia: Learning, Prototyping and Adapting, 2(May), pp. 71–80.

National Building Specification (2015) Uniclass2015_Ro_v1_3. Available at: https://www.thenbs.com/our-tools/uniclass-2015.

National Fire Protection Agency (2013) 'NATIONAL FIRE ALARM AND SIGNALING CODE National Fire Alarm and Signaling Code 2013 Edition', pp. 16–362.

National Institute of Building Sciences (2007) COBie. Available at: https://www.nibs.org/page/bsa_cobie.

Nawari, N. (2012) 'Automating Codes Conformance', Journal of Architectural Engineering, 18, pp. 315–323. doi: 10.1061/(ASCE)AE.1943-5568.0000049.

Nawari, N. O. (2018) Building Information Modeling: Automated Code Checking and Compliance Processes. CRC Press. Available at: https://books.google.pt/books?id=219ltgEACAAJ.

PAS 1192-2 (2013) 'Specification for information management for the capital/delivery phase of construction projects using building information modelling: PAS 1192-2:2013', (1), pp. 1–68. doi: Published by the British Standard Institute. British Standard Limited. ISSN9780580781360. /BIM TASK GROUP.

Preidel, C. and Borrmann, A. (2015) 'Automated code compliance checking based on a visual language and building information modeling', 32nd International Symposium on Automation and Robotics in Construction and Mining: Connected to the Future, Proceedings, (June). doi: 10.22260/isarc2015/0033.

'RIBA-Plan of Work' (2020) Acta Paediatrica, 32(s1), pp. 18–19. doi: 10.1111/j.1651-2227.1944.tb10557.x.

Solibri (2016) Solibri Model Checker, Group, Nemetschek. Available at: https://www.solibri.com/.

Solihin, Wawan and Eastman, C. (2015) 'A Knowledge Representation Approach to Capturing BIM Based Rule Checking Requirements Using Conceptual Graph', CIB W78 Conference 2015.

Solihin, W. and Eastman, C. (2015) 'Classification of rules for automated BIM rule checking development', Automation in Construction, 53, pp. 69–82. doi: 10.1016/j.autcon.2015.03.003.

Succar, B. (2017) '211in Model Uses Table', p. 7. Available at: https://bimexcellence.org/files/211in-Model-Uses-Table.pdf.

Wright, R. N. and Reed, K. A. (1987) 'INTRODUCTION TO SASE: STANDARDS ; 3 ANALYSIS, SYNTHESIS, AND EXPRESSION'.

Young Jr., N. W., Jones, S. a and Bernstein, H. M. (2007) Interoperability in the Construction Industry, SmartMarket Report, Interoperability Issue.

Zhang, H. et al. (2017) 'Semantic web based rule checking of real-world scale BIM models: a pragmatic method', Semantic Web, 0, pp. 1–0. Available at: http://www.semantic-web-journal.net/system/files/swj1621.pdf.

Zhang, J. and El-Gohary, N. (2012) Extraction of Construction Regulatory Requirements from Textual Documents Using Natural Language Processing Techniques, Congress on Computing in Civil Engineering, Proceedings. doi: 10.1061/9780784412343.0057.

LIST OF ACRONYMS AND ABBREVIATIONS

AEC	Architecture, Engineering, and Construction
AI	Artificial Intelligence
AIA	American Institute of Architects
AIR	Asset Information Requirements
ADA	Americans with Disabilities Act
BCA	Singapore: Building and Construction Authority
BCF	BIM Collaboration Format
BEP	BIM Execution Plan
BIM	Building Information Management
bsDD	buildingSMART Data Dictionary
CDE	Common Data Environment
COBie	Construction Operations Building information exchange
CORNET	Construction and Real Estate Network
DMC	Digital Model Checking
EDM	Express Data Manager
EIR	Exchange Information Requirements
FM	Facility Management
GSA	General Service Administration
HSE	Health, Safety, and Environment
HTML	Hypertext Mark-up language
ICC	International Code Council
IDM	Information Delivery Manual
IFC	Industry Foundation Classes
IFD	International Framework For Dictionaries
KPI	Key Performance Indicator
LOD	Level of Definition
LOD	Level of Development
LOG	Level of Geometry
LOI	Level of Information
LOM	Level of Maturity
MVD	Model View Definition
NFPA	National Fire Protection Association
NLP	Natural Language Processing
SASE	Standards Analysis, Synthesis, and Expression
SMC	Solibri Model Checker
SNL	Structured Natural Language
SQL	Structured Query Language
XML	Extensible Mark-up language