# PARAMETRIC MODELLING IN CONSTRUCTION: INVESTIGATING THE QUALITY OF RULE-BASED CHECKING

DENE WARREN
Faculty of Science and Technology, Middlesex University, UK

## ABSTRACT

As thinking moves away from 3D visualisation towards information quality and speed of data exchange, [1], the architectural, engineering and construction (AEC) industry is seeking efficiency via the automation of checking. This study aims to examine the current quality of the checking in design, how it is controlled, and the adoption of automatic checking using rule-based checking software. It is based on the hypotheses that there is a need for automation that will improve quality; a need of a means to achieve automation and a need for understanding in industry. After the investigation of quality in design, this paper poses three questions: first, can a quality system be applied to automatic checking in design?; second, if the potential for rule-based checking is understood and utilised, could quality be assured?; and third, can we identify the optimal balance between manual and automatic checking? The conclusion shows that AEC professionals have low confidence in the quality of design data; the current balance is heavily in favour of manual checking, and while the potential for automatic checking is known, its current utilisation is lacking due to complexity, poor software support and a lack of training. Alarmingly, there is an indication that the majority who deal with automatic checking have no system of validation for new checking rules and only a minority had training in rule authorship.

*Keywords: quality, productivity, rule-based, automatic, checking, parametric, BIM, 3D-model.*

## 1 INTRODUCTION

The development of open-standard object-based 3D modelling based on Industry Foundation Class (IFC) has given rise to the possibility of computer-based interpretable models and the automation of checking based on rules [2]. Structured around a standardised data schema, data can be efficiently checked and cross-checked multiple times for quality. Here quality is defined as conformance to requirements [3]. In contrast to manual checking, automatic checking systems promise consistent, rapid verification of data [4]. They use pre-set rules and applied logic, the quality of which is fundamental to the quality of the finished product. Automated rule checking here is defined as algorithmic software that does not modify a building design but instead assesses a design by the configuration of objects, their relations or attributes [2]. Software such-as Solibri™ Model Checker (SMC) use 50 or so out-of-the-box rules that can be amended or augmented to suit individual needs, but this can be problematic. Expert knowledge is often required to interpret the meaning of regulatory requirements in a particular context with domain-specific assumptions, general knowledge and knowledge of the combinatory effect of related rules [5]. While automatic checking is more efficient, manual checking can ensure an expert review is applied in a domain-specific context and can avoid false positives and multiple negatives. Research has shown, however, that the cognitive challenges of drawing-based checking are such that even experienced professionals are inconsistent [6] and can detect only a small fraction of errors present [7]. Complexity is often underestimated and the application of a quality system by design professionals inconsistent [8]. Clearly, a balance is needed between manual and automatic checking. This study aims to examine the difficulties of using rule-based checking software, the quality of the checking, and how it is controlled.

WIT Transactions on The Built Environment, Vol 192, © 2019 WIT Press
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/BIM190061

## 2 LITERATURE REVIEW

The research into methods of generating computable rules for checking has been broadly divided between two schools of thought; manual logic-based interpretation and language-based interpretation of rules into computable form. The first method is to manually interpret standards and codes and write computer language encoded rules, Eastman et al. [2] and Lee et al. [9] where individual Boolean logic tests are manually written to represent the constraints and test for compliance. The second is Natural Language-Based Interpretation [4] [10] or Natural Language Processing (NLP) technique [11] using semantic web methods to generate rules. In this way, the computer is used to interpret existing standards and codes and automatically derive logic tests to operate on a 3D model to test for compliance. While both processes can generate rules that give automatic alerts due to non-compliance, they still require the intervention of the designer to report and correct the design. The ultimate aim is to automate the system entirely [12].

### 2.1.1 Manual logic interpretation and definitions of rules

This is achieved either by interpretation and translation of written rules or the human language statements into first-order predicate logic [2]. This type of logic functions can be nested together with other rules as 'rule-sets' and can be a powerful means of definition. It is able to return a TRUE or FALSE evaluation but relies on the model being modelled correctly, on the correct application of classification Uniclass 2015 [13] or Omniclass [14], and definitions of building elements defined in the international framework of dictionaries (IFD).

Armstrong [15] characterises the formalising of a set of rules to represent a design process as not straightforward. It is based upon the evaluation of options, both qualitative and quantitive as often there is no one optimal design. Sowa [16] cited by Hjelseth et al. [17] highlights that the process of translation of codes, standards and specifications written for human interpretation is difficult. He calls this the result of 'knowledge soup': overgeneralisation, incomplete definitions, conflicts of defaults and unanticipated applications. Gross [18] emphasised that a lack of understanding in the end-user will lead to a lack of trust of the results and so it is essential to understand where automatic checking ends and manual checking begins. After work in Singapore on CORENET, Eastman et al. [2] warned that results could be compromised both by rule checking algorithm errors and by building model definition errors but the biggest challenge is to detect the false positives, these are rules that simply do not identify the non-compliance. Highlighting the complexity of rule writing Bell et al. [19] proposed a process to generate standardised computable rules based on the principles of knowledge-based engineering. Hjelseth et al. [17] suggested an AEC-based method of development of computable rules. (Section 4.1).

To focus the development of manually interpreted rules Solihin and Eastman [12] made an initial effort at classification by complexity; Class 1 – Rules that require explicit data, Class 2 – Rules that require simple derived attribute values, Class 3 – Rules that require extended data structure that encapsulate higher-level semantic conditions, and Class 4 – Rules that require a 'proof of solution'. The author's suggestion here is that logic complexity is but a subcategory of the primary purpose of functional, presentation and data quality.

While sophisticated checking processes based on manual logic-based Interpretation are successfully deployed, they are recognised as being inflexible and automatic language-based rule interpretation would allow a degree of extensibility so far unknown [20].

### 2.1.2 Natural Language-Based (NLB) interpretation of rules

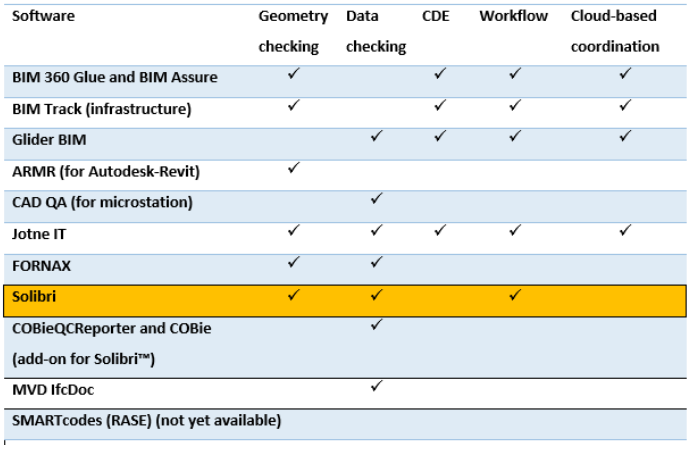
Dikbas et al. [4] proposed that for every code of regulations or standards there is some part of which is simple enough to be easily translated into computer logic. Park et al. [10], introduced a high-level rulemaking language with a sentence-cantered approach. Having an intuitive naming convention directly mapped to the predicate logic of the code sentences made it easier to follow the function and check the logic. This has developed further in the SMARTcodes project [21], a joint project with the International Code Council (ICC), AEC3 Ltd and Digital Alchemy [19]. It enables the translation of building codes and standards from written language rules to computer code, using a dictionary of domain-specific terms being developed as part of the IFD and semi-formal mapping methods [22]. In this way, regulatory codes can be converted into logic rules in a fraction of the time it takes to convert manually, and since the text is so closely associated with the rule logic, it can be considered as a complete quality system. Consequently, rules derived using the NLB technique could revolutionise rule-based checking software, but these are not yet commercially available.

## 3 SOFTWARE

Interoperability is a challenge in the AEC industry which is divided into a discipline and functional specialisms (domains), heavily driven by commercial interests. Technology study estimated that inadequate interoperability led to $15.8 billion in unnecessary annual costs [23]. Software interoperability is defined as the ability for multiple software components written in different programming languages to communicate and interact with another [24]. Often software platforms do not match to the open standard IFC and must be manually mapped. Table 1 gives an overview of a range of currently available checking software and their functions.

Rule-based checking software has developed rapidly over recent years. A Norwegian study [19] found that AEC codes and standards could be accurately represented using manual interpretation and implemented rules using XML or EXPRESS language and this forms the basis of all commercially available rule checking software available today. Checking rules can test geometry (spaces and components) and data (parameters and classification). From the point of view of efficiency, the most useful platform is that which accommodates both geometry, data and workflow checking, and can accept input from models based on

Table 1: Overview of a range of currently available checking software and their functions.



| Software | Geometry checking | Data checking | CDE | Workflow | Cloud-based coordination |
|---|---|---|---|---|---|
| BIM 360 Glue and BIM Assure | ✓ | | ✓ | ✓ | ✓ |
| BIM Track (infrastructure) | ✓ | | ✓ | ✓ | ✓ |
| Glider BIM | | ✓ | ✓ | ✓ | ✓ |
| ARMR (for Autodesk-Revit) | ✓ | | | | |
| CAD QA (for microstation) | | ✓ | | | |
| Jotne IT | ✓ | ✓ | ✓ | ✓ | ✓ |
| FORNAX | ✓ | ✓ | | | |
| Solibri | ✓ | ✓ | | ✓ | |
| COBieQCReporter and COBie (add-on for Solibri™) | | ✓ | | | |
| MVD IfcDoc | | ✓ | | | |
| SMARTcodes (RASE) (not yet available) | | | | | |

buildingSMART [25] open standards – IFC common data schema. The most widely used platform is Solibri™ Model Checker (SMC) and is the focus of this paper. Based on IFC input and its own classification system, SMC provides a library of checking rules for classification and range, pre-checks to pre-defined rules and rules sets for geometric and data checking, concerning information management, space management, and others.

SMC will federate 3D model files and mount onto a viewer for initial screening based on an internal classification system. It is a general check of the model form looking at common errors such as missing information, duplicate components, range, clashes and space validation. Then a detailed geometry and data check can be run using internal rule-sets for conformance to constraints. The power of SMC is that it can check the IFC instance file against any definable rules. SMC provides standard out-of-the-box rules which can be amended or augmented to suit any project or application. The rules are offered as examples and need to be verified for use. Rules do not yet include building regulations and application of safety rules [26], [27], and so limits the scope of automatic checking. Solibri™ have now opened the programming interface (API), for public consumption based on the principle of open standards and it is now possible to add entirely new rules based on JavaScript.

### 3.1.1 Example of logic complexity

Rules can be nested allowing tests to be passed from parent rule to sub-rules to form complex logic checks. As an example of nested logic consider corridor widths specified in Approved Document M, 2015, Provision 3.14, other than dwellings [28]. This provision requires that a corridor has a preferred width of 1.8 m but maybe a minimum of 1.2 m if unobstructed passing spaces 1.8 x 1.8 m square are present. This constraint can be interpreted in one of two ways. The first (a) is to test the width of the corridor is less than 1.8 m wide and greater than 1.2 m wide and has 1.8 m square passing, the second (b) is to test the width is at least 1.2 m and has passing spaces 1.8 m square and is less than 1.8 m wide. Both interpretations are shown diagrammatically in Fig. 1. The equivalent logic yields different results in SMC, see Fig. 2, because the last condition passed is the only one reported. This example is easily corrected within SMC by diligent testing of the nested rule but demonstrates the warnings given by Eastman et al. [2] and Bell et al. [19].

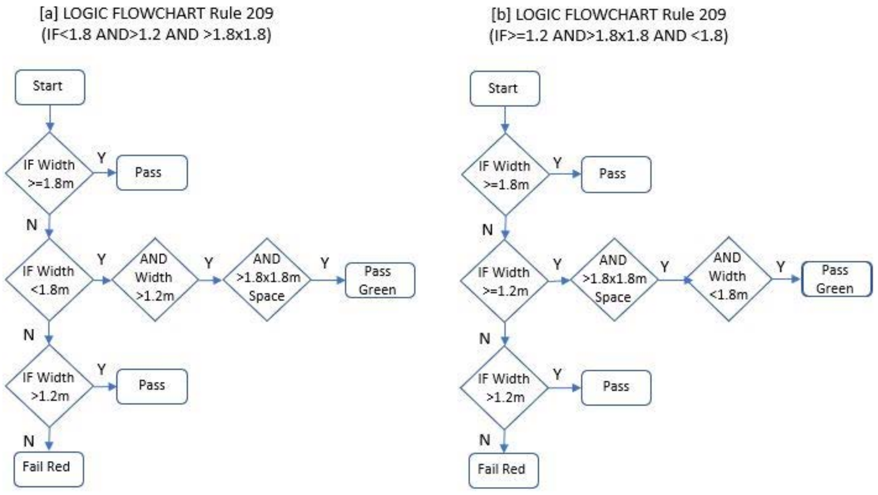
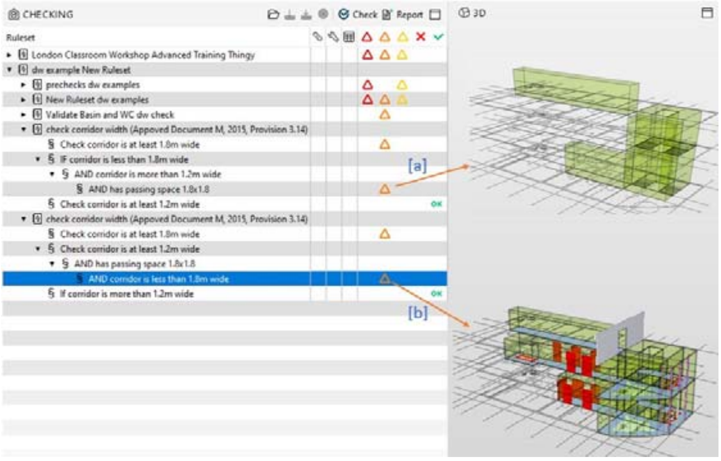


Figure 1: Logic flowchart showing nested rules.



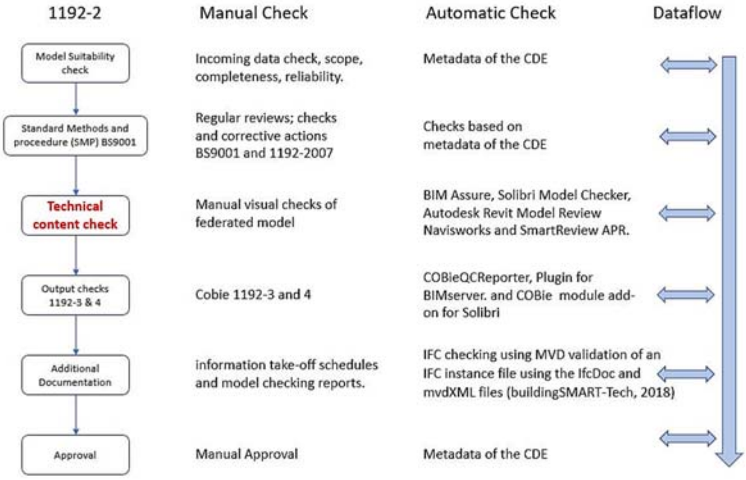
Ruleset

- London Classroom Workshop Advanced Training Thingy
- dw example New Ruleset
  - prechecks dw examples
  - New Ruleset dw examples
  - Validate Basin and WC dw check
  - check corridor width (Appoved Document M, 2015, Provision 3.14)
    - § Check corridor is at least 1.8m wide
    - § IF corridor is less than 1.8m wide
      - § AND corridor is more than 1.2m wide
        - § AND has passing space 1.8x1.8
    - § Check corridor is at least 1.2m wide — OK
  - check corridor width (Appoved Document M, 2015, Provision 3.14)
    - § Check corridor is at least 1.8m wide
    - § Check corridor is at least 1.2m wide
      - § AND has passing space 1.8x1.8
        - § AND corridor is less than 1.8m wide
    - § If corridor is more than 1.2m wide — OK

Figure 2: Screenshot of SMC results.

## 4 QUALITY FRAMEWORK

Formal guidance for a Quality Control Framework is embodied within given PAS1192 [29], BS EN ISO 19650 [30], BS EN ISO 9001 [31] and BS EN ISO 29481 [32]. Where quality control is an activity intended to demonstrate compliance with quality requirements [33], and quality assurance is to demonstrate compliance. The quality framework shown in Fig. 3 is an assemblage of the above, showing the aspects that are currently manual and semi-automatic checks. Semi-automatic checking is provided by discrete operations or executable programmes such as Navisworks, SMC, CobieQCReporter and ifcDoc. The technical content checks are the focus of this paper and whilst there is no formal guidance on authoring rules, guidance is provided by Hjelseth et al. [17], Bell et al. [19] and supported by industry such as Bond Bryan [34].



| 1192-2 | Manual Check | Automatic Check | Dataflow |
|---|---|---|---|
| Model Suitability check | Incoming data check, scope, completeness, reliability. | Metadata of the CDE | |
| Standard Methods and proceedure (SMP) BS9001 | Regular reviews; checks and corrective actions BS9001 and 1192-2007 | Checks based on metadata of the CDE | |
| Technical content check | Manual visual checks of federated model | BIM Assure, Solibri Model Checker, Autodesk Revit Model Review Navisworks and SmartReview APR. | |
| Output checks 1192-3 & 4 | Cobie 1192-3 and 4 | COBieQCReporter, Plugin for BIMserver. and COBie module add-on for Solibri | |
| Additional Documentation | information take-off schedules and model checking reports. | IFC checking using MVD validation of an IFC instance file using the IfcDoc and mvdXML files (buildingSMART-Tech, 2018) | |
| Approval | Manual Approval | Metadata of the CDE | |

Figure 3: Diagrammatic of quality control framework.

### 4.1 Quality system for authoring rules

Hjelseth et al. [17] suggested an AEC-based method of development of computable rules. Based on the principles of knowledge-based engineering, it is noted that each stage is interactive, and loopback is essential for effective application. Bell et al. [19] added certification to seven stages:

Stage I. Define the scope and source of the ruleset. Usually, the standard, code or regulation to be applied

Stage II computability assessment. Rearrange the code so that they are transparent as possible and simplify. Write computable rules. This is best achieved by a domain specialist, not software specialist, because expert knowledge is often required to interpret the meaning of regulatory requirements in a particular context with domain-specific assumptions [5].

Stage III committee assessment approval is a quality assurance of the prepared work from stage II. Teamwork is necessary to identify forgotten steps, limitations of use, and if the rule is necessary. Review to ensure that proposed rules are computable.

Stage IV logic rule notation. Computable rules are transferred to logic notation.

Stage V choose the rule format for presentation of the rules (XML or EXPRESS)

Stage VI implementing (programming) the rules and information text-specific in the rule checking software; this includes the aspects of documentation, changes and testing.

Stage VII Certification. Bell et al. [19] emphasise that Stage I to VI is an iterative process and should be completed by certification.

#### 4.1.1 Automation of the quality framework

An opportunity to automate building information management (BIM) quality framework is given by Dataflow programming [35]. This allows data to flow between discrete operations or executable programmes; implementing dataflow principles of taking the data to the operation rather than the operation to the data. Implementation can either be data-availability-driven (event-driven) or data-request-driven (demand-driven) [35]. Improved efficiency can be depicted as the current flowing in a circuit between electrical components, so data flows between operations in series or parallel. Dataflow programming environments such as LabVIEW [36] and Microsoft Azure [37] are widely used in various industries, but those used in BIM are tailored for model authorship platforms automatic programme interface (API)'s such as Geometry Gym [38] and Grasshopper [39].

#### 4.1.2 BIMSie

An open-source framework is proposed by Aerle [40] that allows creating an 'envelope' for formal, graphical and flexible representations of dataflow processes that can be shared and re-used across projects based on the BIMSie API. The BIM Service interface exchange (BIMSie) [41]. It is designed to connect BIM web services such as BIMQL, mvdXML checker, software authorship platforms and common data environment (CDE)'s in an automated event-driven (data-driven) fashion. The goals of the project are to automate interaction between online BIM services, giving the possibility to innovate with BIM workflow in the 'cloud'.

## 5 CONSULTATION WITH INDUSTRY

Having established that there is guidance for a BIM Quality Framework and that there is software support, although fragmented and complicated, its application is key to investigate the effectiveness and to this end, it is necessary to consult with the AEC industry. Efforts were made to choose from a wide range of roles associated with automatic rule checking, and a combination of a questionnaire, expert interviews and a focus group were used.

Stratified/cluster sampling of the population was chosen from for the questionnaire to include three groups; an IFC's special interest group within LinkedIn social media consisting of 4859 members; technicians and engineers within Mott MacDonald Ltd estimated 3500 employees involved in AEC across the world, and the professional cohort associated with MSc BIMM at Middlesex University consisting of 194 persons. The total population size estimated at 8553. Respondents total number was 92 with a confidence level of 95% and a margin of error of 10%.

Interviews and a focus group were designed to allow inductive reasoning through the qualitative analysis of data collected from experts working within the AEC industry who have experience in BIM. A semi-structured (guided) interview style was adopted, and a seven-step thematic analysis used to draw out conclusions. Consultations consisted of experts of strategy (an online interview with AEC3 Ltd), technical (an online interview with Solibri™) and operational (a focus group held in person with Mott MacDonald Ltd) in October 2018.

It is acknowledged that the small sample size of 1% of the estimated population for the questionnaire limits the reliability of the interpretation to the indication of trends rather than more definitive conclusions and interviews can be dependent on subjective interpretation.

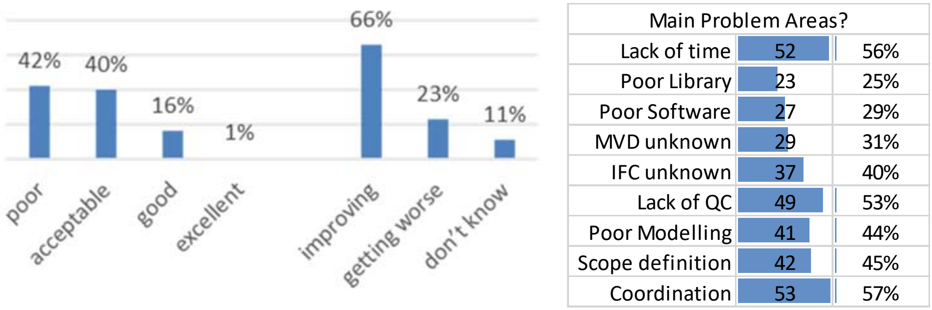
### 5.1 Existing quality (Fig. 4)

Past studies have suggested that AEC professionals have low confidence in the quality of design data [8], and the results of this study continue to support that view. Initial questions asked for an opinion of the general standard of design information in the construction industry. Is it changing? What are the main problem areas? And how can it be improved?

The majority thought the standard of building information is poor and improving. There was no consensus on the reasons for this, but to improve the situation, 70% indicated better training, 61% indicated better automatic checking software and 43% better software support.

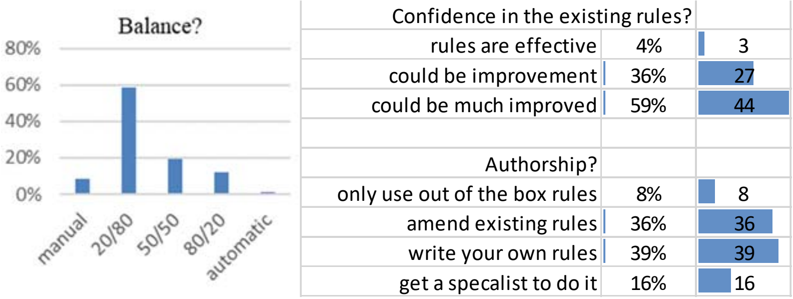
### 5.2 Balance between manual and automatic checking and confidence (Fig. 5)

The questionnaire asked for an estimation of the amount of time spent on manual checking as opposed to automatic checking. If this might change in the future? And if they have confidence in the existing rules or is there room for improvement? 59% of respondents



| Main Problem Areas? | | |
|---|---|---|
| Lack of time | 52 | 56% |
| Poor Library | 23 | 25% |
| Poor Software | 27 | 29% |
| MVD unknown | 29 | 31% |
| IFC unknown | 37 | 40% |
| Lack of QC | 49 | 53% |
| Poor Modelling | 41 | 44% |
| Scope definition | 42 | 45% |
| Coordination | 53 | 57% |

Figure 4: Existing quality results.



| | | |
|---|---|---|
| Confidence in the existing rules? | | |
| rules are effective | 4% | 3 |
| could be improvement | 36% | 27 |
| could be much improved | 59% | 44 |
| | | |
| Authorship? | | |
| only use out of the box rules | 8% | 8 |
| amend existing rules | 36% | 36 |
| write your own rules | 39% | 39 |
| get a specalist to do it | 16% | 16 |

Figure 5: Balance and confidence results.

indicate a balance of checking is 80/20 (manual/automatic), and a majority expect more automatic checking in the future. Results were filtered in accordance with experience, sector, location and role, each indicating that a majority have a checking balance of approximately 80/20 (manual/automatic) so this appears to be common. The majority of respondents to indicate that the rules could be much improved, implying a lack of confidence in the existing provision. 8% said they only use the out-of-the-box rules, 36% amended existing rules, 39% write their own rules and 16% employ a specialist.

### 5.3 Quality system (Fig. 6)

The survey asked respondents if they use a system of validation and approval for amended or newly written rules in checking software? For validation and approval, do you have competent people able to manage the quality of the checking rules?

Respondents indicated that the majority used either a voluntary validation system or had no validation at all. Only a minority had formally trained competent staff for validation and approval of rules, and the majority had gained competence by experience alone.

The results were also separated out in accordance with experience, sector, location and role, and this indicated that the largest group with no validation is the group with experience in the range of 16–25 years', designers, commercial and in the UK.

The survey asked respondents to identify the key area of action that could improve quality. Feedback was spread across all areas with no overall lead. 23% indicated better training, 20% better quality control, 14% more resource, 11% more automatic checking and 11% better modelling skills. Correlation of results shows in Fig. 7 a trend towards the poor and acceptable quality of design information for those who predominantly adopt manual checking and have a voluntary validation of rule authorship in place.

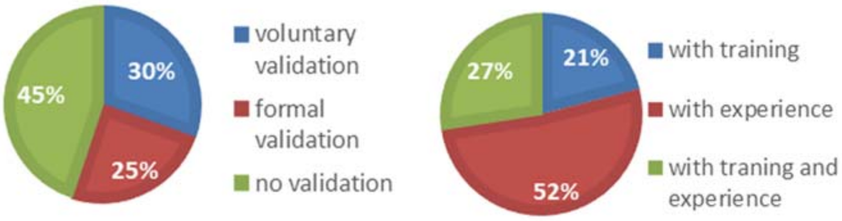


Figure 6: Validation and training results.

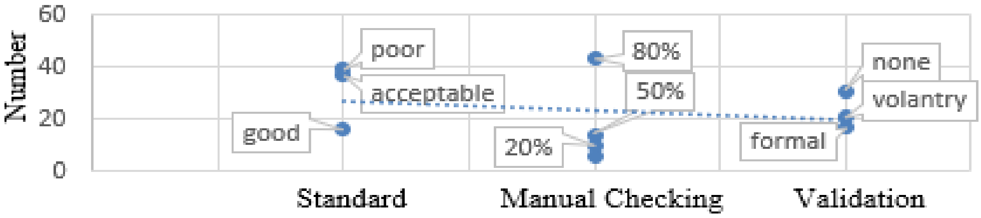


Figure 7: Linear correlation.

### 5.4 Details of respondents and software

The closing questions of the survey asked details of respondents shown in Fig. 8. The survey asked for the identity of the checking software used, and the most popular is 38% Navisworks Manage for geometry checking and 27% Solibri™ for geometry and data.

### 5.5 Focus Group

The Focus Group discussed checking throughout the design process to generate a good quality 3D model. Currently, this involves considerable manual effort from staff at all levels and all process stages. The 3D model is coordinated by designers using Bentley Navigator and data managed with CADQA. However, because of interoperability issues, they are using interlinked spreadsheets and checklists using PowerBI to test data. The workflow system is within ProjectWise and controls verification validation and approval. Competency is assessed internally, and access permissions are regularly reviewed and reported. The focus group agreed that the quality of building information could be improved and that the potential for rule-based checking is not fully realised. They indicated a lack of software support for infrastructure orientated design and the lack of understanding of IFC generally. IFC currently poorly supports infrastructure design with platforms such as BentlyBE, and Revit, which is still operating in IFC2x3. IFC5 which will cater for infrastructure is currently under development, but in the meantime, mapping is required as an IFC Proxy such as buildingSMART MVD for LandXML v1.2 or IFC (Stub) – IFC4x1 Alignment Extension.

The Focus group agreed that automated checking would have a beneficial effect. They estimated that the balance between manual and automatic checking is about 80/20 but with the aspiration of more automation.

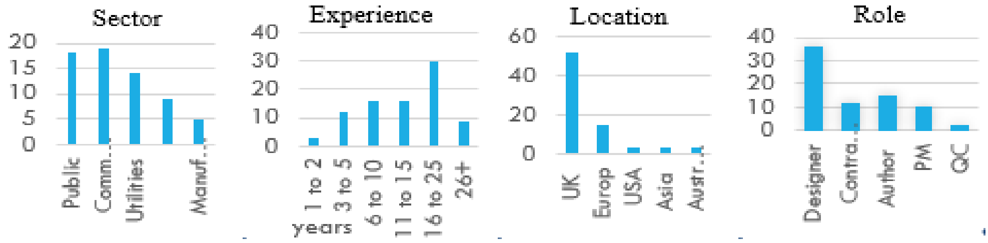


Figure 8: Respondent details.

### 5.6 Expert interviews

The expert interviews agreed that the quality of building design information could be improved and that the potential for rule-based checking is not fully realised. They indicated a lack of understanding of the IFC schema in the industry, a lack of understanding of rules, and rule authoring. Experts also identified poorly constructed models, poor software interoperability and the lack of competent people. Experts felt that checking may swing towards more automatic screening but that some form of a manual check will always be needed. Both expressed an interest in NLB language-based Interpretation for the future development of rules.

## 6 CONCLUSION

Consultation with industry has confirmed the perception of poor quality in design. There is an overreliance on manual checking with only a minority using semi-automatic means, particularly for data. Research has shown that Formal guidance for a Quality Control Framework is in place but does not appear to be implemented effectively. Software is fragmented with poor interoperability issues and poor support for open standards.

Feedback suggested that the potential for semi-automatic rule-based checking is understood but only partly utilised due to complexity of the process, poor software support and the lack of skilled personnel. There appears to be a lack of understanding of rule authoring and in some cases, a lack of a quality framework. The results of the questionnaire indicated Current out-of-the-box rules were being used without a formal validation and rules are being authored without formal training or assessment of capability.

Assurance of quality can be made by an audit of the metadata of the CDE, using semi-automatic open-sourced software. It should include the rule authorship, testing and certification of checking rules such as described by Bell et al. [19], but the feedback suggests that this is far from the case. Training opportunities for automatic checking are needed.

Expert feedback could not identify an optimal balance between manual and automatic checking, suggesting that some manual check will always be required. Increased productivity and quality are inextricably linked to better implementation of automated checking. NLB interpretation of rules, and dataflow programming promise to improve productivity, but better training focused on automatic checking and open standards is required.

## ACKNOWLEDGEMENTS

Thanks to Dr N. Saleeb, Middlesex University for guidance, AEC3 Ltd, Solibri™, Mott MacDonald Ltd who attend interviews and all those in the AEC industry who took the time to complete the questionnaire.

## REFERENCES

[1] Hardin, B. & McCool, D., *BIM and Construction Management: Proven Tools, Methods, and Workflows*, 2nd Kindle ed., Wiley, 2015.

[2] Eastman, C., Lee, J., Jeong, Y. & Lee, J., Automatic rule-based checking of building designs. *Automation in Construction*, **18**, pp. 1011–33, 2009.

[3] Crosby, P., *Quality Without Tears: The Art of Hassle-free Management*, 1st (reprint), McGraw-Hill, 1984.

[4] Dikbas, A., Ergen, E. & Giritli, H., *Managing IT in Construction/Managing Construction for Tomorrow,* CRC Press, 2009.

[5] Solihin, W., *Lessons Learned From Experience of Code-Checking Implementation in Singapore,* 2004.

[6] Fiatech, *Whitepaper FIATECH AutoCodes,* 2012.

[7] Lee, Y.C., Park, J., Won, J., Park, H.K., Uhm, M. & Lee, Y., Can experience overcome the cognitive challenges in drawing-based design review? *16th International Conference on Construction Applications of Virtual Reality*, 2016.

[8] Suermann, P.C. & Issa, R.R.A., Evaluating industry perceptions of Building Information Modeling (BIM) Impact on construction. *Journal of Technology in Construction*, **14**, pp. 574–594, 2007.

[9] Lee, Y.C., Eastman, C.M. & Lee, J.-K., Validations for ensuring the interoperability of data exchange of a building information model. *Automation in Construction*, **58**, pp. 176–195, 2015.

[10] Park, S., Lee, H., Lee, S.-I., Shin, J. & Lee, J.-K., Rule checking method-centered approach to represent building permit requirements. *Proceedings of the 32nd ISARC*, Oulu, Finland, 2015.

[11] Salama, D.M. & El-Gohary, N.M., Semantic modeling for automated compliance checking. *Computing in Civil Engineering*, 2011.

[12] Solihin, W. & Eastman, C., Classification of rules for automated BIM rule checking development. *Automation in Construction*, **53**, pp. 269–82, 2015.

[13] BSI. ISO 12006-2:2015. *Building Construction – Organization of Information About Construction Works – Part 2*: Framework for classification 2015, Online. https://www.iso.org/standard/61753.html. Accessed on: 22 Jan. 2019.

[14] NBIMS. NBIMS-US_V3_2.4_OmniClass_Intro.pdf 2015.

[15] Armstrong, J., *Design Matters: The Organisation and Principles of Engineering Design*, Springer: London, 2007.

[16] Sowa, J.F., *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks/Cole, 2000.

[17] Hjelseth, E., Ergen, E., Giritli, H. & Dikbas, A., *Managing IT in Construction/Managing Construction for Tomorrow: Foundation for Development of Computable Rules*, CRC Press, 2009.

[18] Gross, M.D., Why can't CAD be more like Lego? CKB, a program for building construction kits. *Automation in Construction*, **5**, pp. 285–300, 1996.

[19] Bell, H., Bjørkhaug, L. & Hjelseth, E., *Standardized Computable Rules,* 2009.

[20] Pauwels, P. et al., A semantic rule checking environment for building performance checking. *Automation in Construction*, **20**, pp. 506–18, 2011.

[21] AEC3. SMARTcodes and International Code Council (ICC). International Code Council (ICC) 2012, Online. http://www.aec3.com/en/5/5_013_ICC.htm. Accessed on: 17 Oct. 2018.

[22] Beach, T.H., Kasim, T., Li, H., Nisbet, N. & Rezgui, Y., Towards automated compliance checking in the construction industry. *Database and Expert Systems Applications*, eds H. Decker, L. Lhotská, S. Link, J. Basl & A.M. Tjoa, Berlin: Springer, pp. 366–80, 2013.

[23] Gallaher, M.P., O'Connor, A.C., Dettbarn, J.L. Jr. & Gilday, L.T., Cost Analysis of inadequate interoperability in the U.S. capital facilities industry. *National Institute of Standards and Technology*, **6**(3), 2004.

[24] Wileden, J.C. & Kaplan, A., Software interoperability: Principles and practice. *Proceedings of the (19th) International Conference on Software Engineering*, pp. 631–632, 1997.

[25] buildingSMART. *IFC Industry Foundation Classes. IFC Introduction 2018*, Online. https://www.buildingsmart.org/about/what-is-openbim/ifc-introduction/. Accessed on: 17 Oct. 2018.

[26] Zhang, S., Teizer, J., Lee, J.-K., Eastman, C.M. & Venugopal, M., Building Information Modeling (BIM) and safety: Automatic safety checking of construction models and schedules. *Automation in Construction*, **29**, pp. 183–95, 2013.

[27] Zhong, B.T., Ding, L.Y., Luo, H.B., Zhou, Y., Hu, Y.Z. & Hu, H.M., Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking. *Automation in Construction*, **28**, pp. 58–70, 2012.

[28] HMGov. *Access to and use of Buildings: Approved Document M; Volume 2 – Buildings other than Dwellings*, 2015.

[29] BSI. PAS1192-2:2013 (Incorporating Corrigendum No. 1). *Specification for Information Management for the Capital/Delivery Phase of Construction Projects Using Building Information Modelling,* 2013.

[30] BSI. ISO 19650-1. *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*, 2018.

[31] BSI. BS EN ISO 9001:2015. *Quality Management Systems: Requirements,* London: BSI Group; 2015.

[32] BS EN ISO 29481-1-2017-[2018-08-28--03-02-27 PM].pdf n.d.

[33] PMI. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide),* Kindle ed., Project Management Institute, Inc., 2016.

[34] Bond Bryan. *Testing Open BIM Data Exchange for Client Specific Data Sets*, Bond Bryan BIM. BIM Blog 2018, Online. https://bimblog.bondbryan.co.uk/testing-openbim-data-exchange-for-client-specific-data-sets/. Accessed on: 21 Nov, 2018.

[35] Ayyadurai, C., *Automated Rule Checking for In-House BIM Norms of Building Models,* Eindhoven University of Technology, Department of the Build Environment Design Systems Group, 2016.

[36] LabVIEW: National Instruments. *What is LabVIEW?* 2018, Online. http://www.ni.com/en-gb/shop/labview.html. Accessed on: 12 Nov. 2018.

[37] Microsoft Azure. *Microsoft Azure Cloud Computing Platform & Services*, 2018. https://azure.microsoft.com/en-us/. Accessed on 12 Nov. 2018.

[38] Grasshopper. *Geometry Gym for BIM*, 2018, Online. https://www.grasshopper3d.com/groups/group/show?groupUrl=geometrygym. Accessed on: 12 Nov. 2018.

[39] Grasshopper3d. Grasshopper, *Algorithmic Modeling for Rhino* 2018. Online. https://www.grasshopper3d.com/events/graph-modelling-with-grasshopper3d-introduction-to-data-flow. Accessed on: 12 Nov. 2018.

[40] Aerle, M.L., *IFC Dataflow – An Open Framework for Connecting BIM Web Services,* Eindhoven University of Technology, Department of the Build Environment Design Systems Group, 2015.

[41] NIBS, buildingSMART alliance, *BIM Service Interface Exchange (BIMSie) Project,* National Institute of Building Sciences, 2015, Online. https://www.nibs.org/page/bsa_bimsie. Accessed 11 Nov. 2018.