Soliman-Junior, J., Tzortzopoulos, P., and Kagioglou, M. 2020. "Exploring Mistakeproofing in Healthcare Design." In: Tommelein, I.D. and Daniel, E. (eds.). *Proc. 28th Annual Conference of the International Group for Lean Construction (IGLC28),* Berkeley, California, USA, doi.org/10.24928/2020/0034, online at iglc.net.

EXPLORING MISTAKEPROOFING IN HEALTHCARE DESIGN

Joao Soliman-Junior[1], Patricia Tzortzopoulos[2], and Mike Kagioglou[3]

ABSTRACT

The process of verifying the compliance of design solutions to regulations is critical in healthcare design due to the complexity of the requirements present in healthcare settings. The majority of previews research concentrates on "mistake-searching", assuming that design will not be compliant and, as such, necessitates an approach focussed on finding inconsistencies, rather than avoiding mistakes during the design process. This paper reports findings of an ongoing research that follows the Design Science Research approach, with the aim of exploring how existing technologies can support incorporating mistakeproofing (*poka yoke*) into healthcare design, framed within the regulations compliance process. A set of technological strategies was assessed according to mistakeproofing principles. The analysis evidenced their characteristics, benefits, limitations and examples of application. These technologies rely on the use of hybrid approaches, providing assistance to designers across all design stages, which in turn supports better decision-making and contributes towards improving value generation. A theoretical framework was proposed based on the synergy between design support systems, requirements subjectivity and jidoka. It highlights the importance of better understanding and enhancing the relationship between human designers and different technologies through automation.

KEYWORDS

Mistakeproofing (mistake-proofing, mistake proofing), *poka yoke*, healthcare design, regulations, autonomation.

INTRODUCTION

Many sources of information support decision-making across the design process. These are extensive sets of data (Kiviniemi and Fischer 2004), representing a basic framework upon which solutions are created and assessed. Regulatory information from codes, standards and other documents are responsible for a considerable part of this data (Macit İlal and Günaydın 2017), introducing different types of constraints into the design process (Von der Tann et al. 2018). Within healthcare design, this becomes even more critical due to (i) the complexity of regulatory documents associated to this type of building, describing their functional and technological aspects (Chellappa 2009); and (ii) the impact

[1] PhD Student, School of Art, Design and Architecture, University of Huddersfield, Huddersfield, HD1 3DH, UK, Joao.SolimanJunior@hud.ac.uk, orcid.org/0000-0002-8089-8628

[2] Professor, School of Art, Design and Architecture, University of Huddersfield, Huddersfield, HD1 3DH, UK, P.Tzortzopoulos@hud.ac.uk, orcid.org/0000-0002-8740-6753

[3] Professor, School of Engineering, Western Sydney University, Penrith NSW 2751, Australia, M.Kagioglou@westernsydney.edu.au, orcid.org/0000-0003-3521-1484

of design quality on services delivered in these facilities, which, in turn, have a major influence on health outcomes (Tzortzopoulos et al. 2005).

Assessing design proposals by verifying the fulfilment of requirements is an important step in the design process, which provides an opportunity to improve design value (Fiksel and Hayes-Roth 1993). Past research has extensively explored the use of automated rule checking to support design assessment (Eastman et al. 2009; Nawari 2019). This was done by framing their scope within the compliance process, considering different regulatory frameworks (Macit İlal and Günaydın 2017; Nawari 2019). Most of these initiatives emerged from a fragmented background, related to the proposition of different sets and types of solutions to overcome limitations of manual-based compliance approaches (Eastman et al. 2009; Macit İlal and Günaydın 2017).

Past research on automated rule checking indicate a move towards a digital and technology-based assessment process (Soliman-Junior et al. 2019). Despite representing an advancement, most efforts still emulate the common rationale of "mistake-searching". They start from the assumption that design proposals will not be compliant, and mistakes (or non-compliances, in this context) need to be detected, flagged and informed to the design teams, so they can react and modify the design accordingly (i.e. rework). This is the main issue associated to such underlying reactive reasoning – finding inconsistencies after they are made, rather than aiming to avoid design mistakes in the first place.

This paper presents a theoretical analysis exploring how existing technologies can support incorporating mistakeproofing [4] (*poka yoke*) into healthcare design. The assumption is that embedding mistakeproofing into design operations can contribute towards shifting assessment to a more iterative, simultaneous and continuous design-checking process, reducing rework. This contributes to improve design outputs in terms of compliance to regulatory requirements and therefore acts as an assurance tool rather than a testing tool. It is important to highlight that in this paper, mistakes are considered to be any design non-compliance to the applicable regulatory framework. Thus, design mistakes represent, in this context, a potential source of derogation and rework from both the perspective of design teams and regulatory institutions in charge of assessing design proposals.

DESIGN ASSESSMENT THROUGH HUMAN-TECHNOLOGY PERSPECTIVE

There are two different understandings of how automated rule checking might support the design compliance process (Hjelseth 2016). These are (i) *compliance checking* seen as a separate process from the overall design, performed at determined times (usually done by importing the building model into a rule-checking engine); and (ii) *design solution checking*, usually performed in a continuous way, as a support and parallel operation during the design process. Lee et al. (2019) concluded that the second understanding supports design optimisation by correcting errors and exploring different design alternatives. The main limitation of this study (Lee et al. 2019) was incorporating information from subjective requirements, which still remains a challenge according to the authors perspective.

In fact, the problematic rhetoric around using automation in design assessment and requirements subjectivity is recognised by the literature as a major shortfall (Nawari 2012; Dimyadi and Amor 2013; Lee et al. 2019). This is because human involvement is

[4] Mistakeproofing is spelled in one word in this paper for consistency with other IGLC authors.

fundamental in dealing with the abstract and subjective elements, which are implicitly embedded within regulatory documents (Nawari 2012; Solihin and Eastman 2016). Technology helps human activities during design in different ways (Heumann and Davis 2019), as well as using different degrees of automation (Soliman-Junior et al. 2019). In practice, this indicates a need for hybrid approaches, which support design assessment through better exploration of human-technology interactions. Such approaches cope with requirements subjectivity in such a way as to not hinder the creativity of human designers during the design process – which is imperative to increase design value.

These approaches incorporate a well-known concept within the Lean community, originated in manufacturing i.e. jidoka or autonomation (Shingo 1986), “automation with a human touch” (Liker 2004). Autonomation is related to machines that were designed so they could identify problems and promptly stop production. The aim was freeing people to use their creativity and unique reasoning for essential operations, related to increasing value (Liker and Meier 2006). As such, autonomation is fundamentally related to (i) built-in quality; (ii) mistakeproofing; and (iii) assuring that humans are free to perform value-adding work, i.e. people are in the centre of the system (Liker, 2004). These principles are explored here, applied to design, by investigating different digital technologies and their potential applications from an autonomation perspective.

MISTAKEPROOFING

Mistakeproofing is also known by the Lean community as *poka yoke*. It was originally defined in manufacturing by Shingo (1986), and relates to the development and application of creative devices that can effectively reduce the chance of making errors, so waste can be eliminated (Liker 2004). According to McMahon (2016), mistakeproofing relies on processes bounded by constraints, aiming to prevent incorrect operations. ASQ (2019) complements the previous definition by stating that using automated techniques to avoid errors or identifying errors promptly is at the centre of mistakeproofing. This is fundamentally a paraphrase of Shingo’s own perception (Shingo 1986), who claimed the importance of quickly finding mistakes is to avoid rework and other types of waste, due to late detection of errors. The concept of mistakeproofing is based on six principles (Shingo 1986; McMahon 2016; Tommelein 2019), which are presented in Table 1.

The creative and subjective nature of the design process is a fruitful setting for the occurrence of inconsistencies and errors. These errors are unintentional and stem from the idiosyncratic way humans think and undertake tasks (Tommelein 2019), associated to cognitive error mechanisms (Stewart and Grout 2001). In design, the negative effects of mistakes must be alleviated, so they do not compromise further decision-making and reduce their impacts on the project.

Mistakeproofing has been an area of research poorly explored over the last decades, due to the lack of systematic use and studies in the construction industry (Tommelein 2019). Studies related to its application focussed on building design are even more scarce. Examples of using mistakeproofing in design have only been presented by Tommelein (2008; 2019), amongst an extensive set of applications from different domains. There is still no discussion on how mistakeproofing could be supported by technologies used in the design process.

Even though mistakeproofing was defined within the manufacturing domain, its six principles are also valid to the design of a product and its associated operations (Tommelein 2019). In fact, these principles can be understood as categories of mistakeproofing, organised by their effectiveness in reducing waste. They might support identifying how mistakeproofing can be incorporated to different operations across the design process, considering the regulatory framework as a baseline.

Table 1: Principles of Mistakeproofing
(Shingo 1986; McMahon 2016; Tommelein 2019)

Principle	Definition
Elimination	Removing the possibility of error by redesigning the product or operation in such a way as the step of the operation or the part of the product becomes unneeded.
Prevention	Designing a product or operation in such a way as to prevent the occurrence of any mistake at all.
Replacement	Substitute an operation by another which is more reliable, with the aim of improving repeatability and improving consistency.
Facilitation	Making operations easier by using various techniques so they are less error-prone.
Detection	Identifying a mistake straightaway so it can be corrected before further processing.
Mitigation	Minimizing the effects and impact of errors.

RESEARCH METHOD

This paper reports a theoretical analysis based on findings from an ongoing research, which is framed within the healthcare design and compliance context. This research adopts the Design Science Research approach (DSR), which supports solving problems that have both practical and theoretical relevance, in an effective and innovative way (Lukka 2003). By using DSR, the research problem is understood while the artefact is produced (Holmström et al. 2009). This process consists of iterative cycles of analysis, understanding, development and refinement. Thus, findings presented by this paper do not represent the artefact on their own, but constitute part of the reasoning behind this evolving process.

The main sources of evidence in this paper are related to theoretical data from literature review, by providing fundamental inputs on (i) automation and technology applied to design; and (ii) clarifying concepts and principles related to autonomation and mistakeproofing. Based on literature, different technological strategies were identified and explored under the six principles of mistakeproofing. The analysis undertaken by this study is also partially informed by empirical data which is not presented in the paper. These correspond to a study developed in collaboration with an institution responsible for Primary Healthcare buildings across England, and relate to: (i) analysis of regulations, which supported understanding the content of information from requirements and how mistakeproofing could be used under different strategies; and (ii) meetings and interviews, which suggested how design support systems can naturally incorporate mistakeproofing, through the perspective of different designers.

TECHNOLOGIES FOR DESIGN SUPPORT SYSTEMS BASED ON MISTAKEPROOFING

The use of mistakeproofing can be explored in healthcare design by analysing how existing technologies relate to its six principles. These also outline strategies to support the development of design support systems. The analysis presented by Figure 1 represents the key findings of this paper, which are discussed in this section. It is important to acknowledge that some of these technologies already exist and are used in practice, while others still demand some level of development to be further incorporated in such design support systems, as indicated by the same figure.

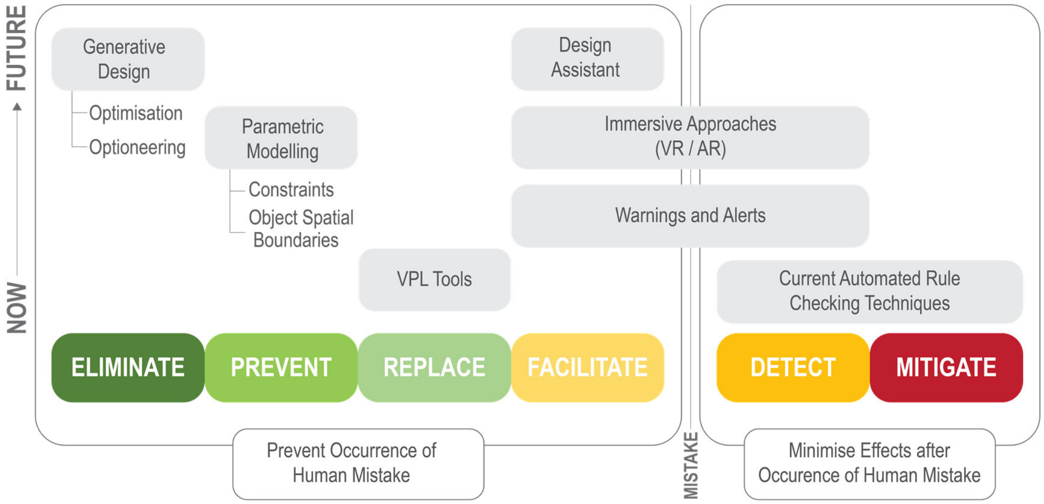

FUTURE
NOW
Generative Design
Optimisation
Optioneering
Parametric Modelling
Constraints
Object Spatial Boundaries
VPL Tools
Design Assistant
Immersive Approaches (VR / AR)
Warnings and Alerts
Current Automated Rule Checking Techniques
ELIMINATE
PREVENT
REPLACE
FACILITATE
DETECT
MITIGATE
Prevent Occurrence of Human Mistake
MISTAKE
Minimise Effects after Occurence of Human Mistake

Figure 1: Technologies for design support systems and their relationship to the mistakeproofing principles

GENERATIVE DESIGN

Generative design, also defined as algorithmic design, is related to "virtual geometric processes that are highly numerically controlled and constrained parametrically" (Garber 2014, p. 125). In fact, the use of generative design, by applying Artificial Intelligence (AI), has been explored since very early stages of digital design. There are two different types of approaches related to using AI-based generative design: design optimisation and design optioneering. Design optimisation is associated to genetic algorithms, and its use aims to indicate the most suitable solution (Arora 1986), by solving problems with multiple variables and disciplines (Gerber et al. 2012). Design optioneering, on the other hand, takes advantage of such algorithms to generate multiple design alternatives (Gerber et al. 2012), which can support further decision-making. While the first approach provides the best possible solution, based on a pre-defined set of criteria; the second provides many functional solutions, so designers can choose the most suitable one, according to their own thoughts and criteria. These approaches could support incorporating mistakeproofing by **eliminating** the possibility of error, while human designers were to solve the same problems; e.g. sorting layout of floorplans based on constraint requirements of area and spatial adjacencies. In both cases, the possibility of eliminating mistakes relies on whether the set of rules and constraints from AI algorithms have been properly defined. Moreover, the AI-based interfaces must be aligned with the designers' workflow and their intuitive reasoning (Heumann and Davis 2019).

PARAMETRIC MODELLING

Parametric modelling supports defining a system by describing its parameters and relationships (Kensek and Noble 2014). Its use can **prevent** the occurrence of errors by applying constraints and object spatial boundaries, within object-based modelling tools. Constraints are already used to support modelling operations in current software applications, such as Autodesk Revit and Graphisoft Archicad. These are encoded within software systems to force intent while designing (Kensek and Noble 2014). They reflect relationships between objects according to real limitations e.g. a window object can only be inserted if it is associated to a host object type wall. Object spatial boundaries are related to the space required by objects in use. They correspond to their projection and associated space for manoeuvre. This approach incorporates a specific type of information from healthcare regulatory requirements, associated to free spaces around specific objects, e.g. minimum free distances in front and on both sides of beds. The use of parametric modelling could be further expanded, by modelling constraints and using them as a way of embedding information from the regulatory framework into object families and building models.

VISUAL PROGRAMMING LANGUAGE (VPL) TOOLS

Visual programming language (VPL) tools, such as Dynamo and Grasshopper, could be used to **replace** part of the design process, typically done by humans under object-based modelling interfaces. These programming languages use graphical elements to support creating logic algorithms (Kensek and Noble 2014). One of the advantages of using VPL approaches is the fact they can be maintained without much specialized support (Ghannad et al. 2019). Moreover, they are user-centred, providing enough flexibility to be adapted to different operations and designers (Preidel and Borrmann 2015, 2016). The use of VPL tools in design support systems can be associated to prone to error activities related to repetitive operations, e.g. including the same object many times in the building model and adapting different elements within the building model, according to fixed parameters and constraints.

WARNINGS AND ALERTS

Using warnings and alerts in object-based modelling interfaces could incorporate mistakeproofing in the design process by: (i) **facilitation** – visually flagging potential mistakes before they are made; or (ii) **detection** – promptly informing designers right after any mistake is detected. While the first approach relies on warning signals, which visually alert designers about potential mistakes, so they could react and proceed straightaway with the design process; the second is related to identifying and informing designers about any mistakes already made (Mathot et al. 2019). It is important to acknowledge the dichotomy related to warnings and alerts on mistakeproofing, since they could be either used before or after the occurrence of mistakes. They also could force a complete stop in the design process, by introducing compulsory constraints in the software algorithm, so errors could be corrected before advancing further – which is at the core of mistakeproofing (Shingo 1986).

IMMERSIVE APPROACHES (VR/AR)

The use of immersive approaches through Virtual and Augmented Reality (VR/AR) is not a novelty on their own in the AEC context, but they could be adopted differently towards better incorporating mistakeproofing principles. Immersive approaches **facilitate** designers' work, but also support quick detection of mistakes. They rely on a concept called manual-assist collaboration, which uses robots and digital tools to excel humans' capabilities, under a continuous and integrated way (Johns et al. 2019). They enable evidencing contextual information about the product and the associated processes. Thus, they could provide an immersive environment, helping human designers to perform their operations in all stages of design (Betti et al. 2019).

DESIGN ASSISTANT

A design assistant could **facilitate** the design process by supporting designer's decision-making. Within the context of design support systems, a design assistant does not exist yet and should be envisioned as a constant guide. It could inform designers on their decision-making, based on information from the regulatory framework. It should provide the right information (what is needed) at the exact time designers need to use it (when is needed), by suggesting where, how and why elements should be inserted across the design process. Moreover, it could track decision-making by also recording where its evidence or knowledge was needed and how they influenced and modified the design, promoting continuous improvement. This type of approach demands interfaces between humans and technologies to be excelled up to their current limits, since such integration relies on maintaining designers' workflows and cognitive processes (Heumann and Davis 2019). The adoption of this strategy to object-based modelling tools, can, ultimately, provide a platform for continuous learning from potential mistakes, improving the design process by also informing designers about what could have gone wrong.

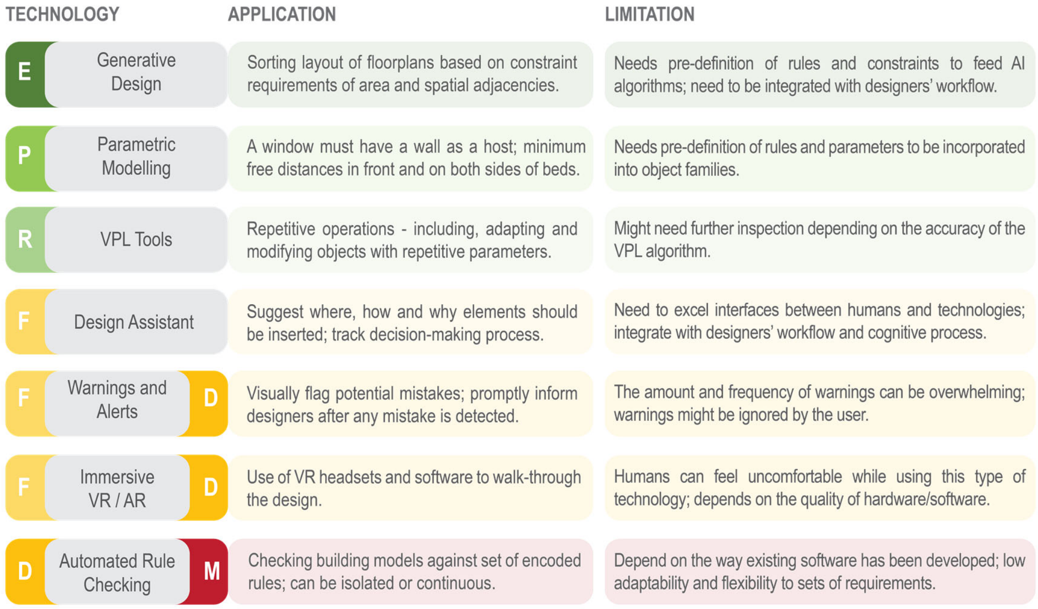
CURRENT AUTOMATED RULE CHECKING TECHNIQUES

Despite the advancements reported by literature over the last years, the use of automated rule checking to support the design assessment is not widely observed in practice. There are many limitations on existing software, mostly related to the way these were developed, impacting their adaptability and flexibility to incorporate information from different regulatory contexts (Eastman et al. 2009; Nawari 2019). In situations where automated rule checking techniques are used, they relate to the lowest levels of mistakeproofing, in which errors and inconsistencies have already happened. They **detect** and **mitigate** the negative effects of mistakes in the design, depending on how their associated processes and workflows are structured, as discussed by Hjelseth (2016): (i) if automated rule checking is used at the end of the design stages, under an isolated and non-repetitive approach, it relates to the mitigation principle; on the other hand, (ii) if automated rule checking is incorporated to design at different stages, and performed under an iterative and continuous way, it reflects the detection principle. In the first case, the effects of mistakes can only be mitigated. Therefore, they impact on other design decisions, potentially affecting the overall course of the associated project. Moreover, current uses of automated rule checking demand building models to be exported from their native object-oriented tool to other applications, under different file formats. This can eventually result in and another error-prone situation, due missing information and conflicting files.

DISCUSSION

Considering the healthcare design context and the associated regulatory framework, most of the strategies presented in the previous section could be introduced to operations through hybrid approaches. They could further support reducing the occurrence of non-compliances in later stages of the design process. In fact, the practical adoption of these technological strategies could reduce, and potentially eliminate derogations[5] from the healthcare design process. They are one of the main causes of rework and clashes from a design perspective; but also, are responsible for many delays, overspendings and rejections from planning and funding organisations, under a project perspective.

The application of technology as discussed above, often represent an ideal situation. There are many limitations associated to the way they could be incorporated to design operations, which still demand further exploration in practice. Figure 2 presents an overview of these technologies, examples of their application and their main limitations identified in this study. In order to enable mistakeproofing, many of the proposed strategies rely on the success of prior steps of the digital design process. This is associated to precise definition of parameters, algorithms and constraints, based on information associated to the regulatory framework.



	TECHNOLOGY		APPLICATION	LIMITATION
E	Generative Design		Sorting layout of floorplans based on constraint requirements of area and spatial adjacencies.	Needs pre-definition of rules and constraints to feed AI algorithms; need to be integrated with designers' workflow.
P	Parametric Modelling		A window must have a wall as a host; minimum free distances in front and on both sides of beds.	Needs pre-definition of rules and parameters to be incorporated into object families.
R	VPL Tools		Repetitive operations - including, adapting and modifying objects with repetitive parameters.	Might need further inspection depending on the accuracy of the VPL algorithm.
F	Design Assistant		Suggest where, how and why elements should be inserted; track decision-making process.	Need to excel interfaces between humans and technologies; integrate with designers' workflow and cognitive process.
F	Warnings and Alerts	D	Visually flag potential mistakes; promptly inform designers after any mistake is detected.	The amount and frequency of warnings can be overwhelming; warnings might be ignored by the user.
F	Immersive VR / AR	D	Use of VR headsets and software to walk-through the design.	Humans can feel uncomfortable while using this type of technology; depends on the quality of hardware/software.
D	Automated Rule Checking	M	Checking building models against set of encoded rules; can be isolated or continuous.	Depend on the way existing software has been developed; low adaptability and flexibility to sets of requirements.

Figure 2: Application and limitation of technologies under a mistakeproofing perspective

From the analysis carried out in this paper, an important characteristic of applying mistakeproofing in design was observed. In design, mistakeproofing strategies often do not fit exclusively into one principle. This is due to the different ways the same technology can be used, according to different factors such as context, workflow and the unique reasoning of every designer. Moreover, it can be difficult to determine what is a mistake in design and what is the relationship between mistakes and rework in this process. Differently from production, design is based on iterative and complex operations. Thus, much of the potential rework can actually be the source of emergent and creative solutions.

[5] In the UK healthcare context, derogations are all design deviations from the standards, regulations and other guidance documents. They emerge from circumstances where it may be impossible or impractical to attain these requirements. Derogations are often presented and documented by a schedule, which states the requirements being derogated, the reason why design is not compliant, the associated risks and how their impact shall be mitigated.

This understanding suggests that mistakeproofing can assume different perspectives within the production and design domains, as evidenced by this dichotomy.

Finally, it is also important to observe holistically how the strategies previously explored embrace *jidoka* in design, at the same time they address key issues around requirements subjectivity, from the automated rule checking research. Figure 3 demonstrates how the technological strategies (represented by design support systems) can work, under a theoretical framework, as a convergent point between Lean and the issue around requirements subjectivity. It can be understood as a synergetic element between (i) the subjectivity embedded in the regulatory framework, potentially addressed through hybrid solutions; and (ii) *jidoka*, through mistakeproofing. In both streams, this synergy relies on the interaction between humans and technologies.

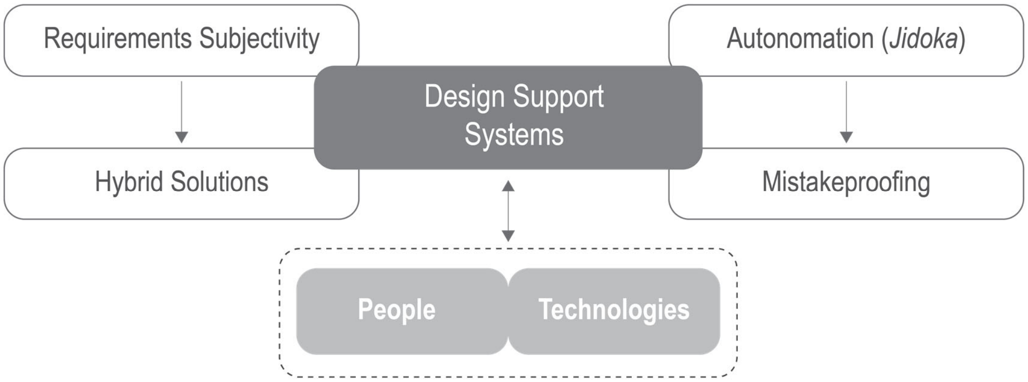


Figure 3: Relationship between requirements subjectivity and jidoka through mistakeproofing, by using hybrid solutions in developing design support systems

CONCLUSIONS

This paper explored mistakeproofing within the design domain, under a theoretical perspective. The aim of this study was to explore how existing technologies could support incorporating mistakeproofing into healthcare design. This was done by investigating relationships between different technological strategies and mistakeproofing principles to support design operations. The main limitation of the analysis carried out in this paper is that technologies were mostly assessed under a theoretical perspective. Thus, they have not been applied and validated in design practice. These technologies have been individually presented, identifying their main characteristics, benefits, limitations and examples of application. By doing so, this paper demonstrated the potential feasibility of using mistakeproofing within the design domain.

Findings presented in this paper also evidenced the need for hybrid approaches in design. The use of technology should be understood as a means to provide continuous advice and assistance across all stages and operations of the design process. A theoretical framework emerged from the synergy observed between design support systems, requirements subjectivity and *jidoka* (Figure 3). The interaction between people and technologies is the foundation of this synergy, reflecting the basic definition of *jidoka*. Future work should further investigate this framework and test the theoretical contribution from each of the technologies in practice. Findings from this study suggest that exploring mistakeproofing in the design domain relies on better understanding the relationship between human designers and technologies – and how they are enablers of one another.

ACKNOWLEDGMENTS

The authors would like to thank the Innovative Design Lab (IDL), University of Huddersfield. Moreover, we would like to thank Community Health Partnerships (CHP) collaborators, project and design team members, for their time, support and the opportunity to help in this investigation.

REFERENCES

Arora, J.S. and Baenziger, G. (1986). "Uses of artificial intelligence in design optimization." *Computer Methods in Applied Mechanics and Engineering*, 54(3), 303-323.

ASQ (2019). https://asq.org/quality-resources/mistake-proofing. Access on 02 Dec. 2019.

Betti, G., Aziz, S., and Ron, G. (2019). "Pop-Up Factory: Mixed Reality Installation for the MakeCity Festival 2018 in Berlin." In: *Design Modelling Symposium Berlin* (pp. 265-276), Springer, Cham.

Chellappa, J.R. (2009). "BIM + healthcare: utilization of BIM in the design of a primary healthcare project." University of Hawai'i, USA.

Dimyadi, J. and Amor, R. (2013). "Automated Building Code Compliance Checking – Where is it at?" P*roceedings of CIB WBC 2013*, 172-185.

Eastman, C., Lee, J., Jeong, Y., and Lee, J. (2009). "Automatic rule-based checking of building designs." *Automation in Construction*, 18(8), 1011-1033.

Fiksel, J., and Hayes-Roth, F. (1993). "Computer-aided Requirements Management." *Concurrent Engineering,* 1(2), 83-92.

Garber, R. (2014). *BIM Design: Realising the Creative Potential of Building Information Modelling.* AD Smart, John Wiley & Sons, Inc., Chichester, West Sussex, United Kingdom.

Gerber, D.J., Lin, S.H., Pan, B., and Solmaz, A.S. (2012). "Design optioneering: multi-disciplinary design optimization through parameterization, domain integration and automation of a genetic algorithm." In: *Proceedings of the 2012 Symposium on Simulation for Architecture and Urban Design*, pp. 1-8.

Ghannad, P., Lee, Y. C., Dimyadi, J., and Solihin, W. (2019). "Automated BIM data validation integrating open-standard schema with visual programming language." Advanced Engineering Informatics, Elsevier, 40(January), 14-28.

Heumann, A., and Davis, D. (2019). "Humanizing architectural automation: a case study in office layouts." *Design Modelling Symposium Berlin*, Springer, 662-670.

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

Holmström, J., Ketokivi, M., and Hameri, A.-P. (2009). "Bridging Practice and Theory : A Design Science Approach." *Decision Science*, 40(1), 65-87.

Johns, R. L., Anderson, J., and Kilian, A. (2019). "Robo-Stim: Modes of Human Robot Collaboration for Design Exploration." *Design Modelling Symposium Berlin* (pp. 671-684). Springer, Cham.

Kensek, K. and Noble, D., (2014). *Building Information Modeling: BIM in Current and Future Practice.* Wiley, Hoboken, New Jersey, USA. 432 pages.

Kiviniemi, A., and Fischer, M. (2004). "Requirements Management Interface to Building Product Models." 1-12.

Lee, P.-C., Lo, T.-P., Tian, M.-Y., and Long, D. (2019). "An Efficient Design Support System based on Automatic Rule Checking and Case-based Reasoning." *KSCE Journal of Civil Engineering*, 23(5), 1952-1962.

Liker, J.K. (2004). *The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer*. McGraw-Hill Education, Madison, WI.

Liker, J.K. and Meier, D. (2006). *The Toyota Way Fieldbook: A Practical Guide for Implementing Toyota's 4Ps*. New York, London: McGraw-Hill.

Lukka, K. (2003). "The constructive research approach." *Case study research in logistics*. Publications of the Turku School of Economics and Business Administration, Series B, 1(2003), 83-101.

Macit İlal, S., and Günaydın, H. M. (2017). "Computer representation of building codes for automated compliance checking." *Automation in Construction*, 82(June), 43-58.

Mathot, M., Hohrath, B., Rolvink, A., and Coenders, J. (2019). "Design Modelling with Next Generation Parametric System Packhunt." *Design Modelling Symposium Berlin* (pp. 277-287). Springer, Cham.

McMahon, T. (2016). "Six Principles of Mistake Proofing." *A Lean Journey*, 20 July 2016, www.aleanjourney.com/2016/07/six-principles-of-mistake-proofing.html. Access on 02 Dec. 2019.

Nawari, N.O. (2012). "The Challenge of Computerizing Building Codes in a BIM Environment." *Computing in Civil Engineering*, 1, 285-292.

Nawari, N. O. (2019). "A Generalized Adaptive Framework (GAF) for Automating Code Compliance Checking." *Buildings*, 9(4), 86.

Preidel, C., and Borrmann, A. (2015). "Automated Code Compliance Checking Based on a Visual Language and Building Information Modeling." *Proc. 32nd International Symposium of Automation and Robotics in Construction*, 15-18 June, 256-263.

Preidel, C., and Borrmann, A. (2016). "Towards code compliance checking on the basis of a visual programming language." *Journal of Information Technology in Construction*, 21(July), 402-421.

Shingo, S. (1986). *Zero Quality Control: Source Inspection and the Poka-Yoke System*. Portland, Oregon: Productivity Press.

Solihin, W., and Eastman, C. M. (2016). "A knowledge representation approach in BIM rule requirement analysis using the conceptual graph." *Journal of Information Technology in Construction*, 21(March), 370-401.

Soliman Junior, J., Parise Baldauf, J., Tzortzopoulos, P., Formoso, C. T., and Kagioglou, M. (2019). "The role of building information modelling on assessing healthcare design." *Constructing Smart Cities: Proc. 22nd CIB World Building Congress (CIB2019)*, 17-21 June, The Hong Kong Polytechnic University, Hong Kong, China.

Stewart, D. M., and Grout, J. R. (2001). "The human side of mistake-proofing." *Production and Operations Management*, 10(4), 440-459.

Von der Tann, L., Angelino, M., Crick, R., and Taylor, C. (2018). "Rethinking Design Standards as Learning Frameworks." *ICIF White Paper Collection*, International Centre for Infrastructure Futures (ICIF), London, UK.

Tommelein, I.D. (2008). "'Poka Yoke' or Quality by Mistake Proofing Design and Construction Systems." *In Proc. 16th Annual Conference of the International Group for Lean Construction (IGLC 27)*, Manchester, UK.

Tommelein, I.D. (2019). "Principles of Mistakeproofing and Inventive Problem Solving (TRIZ)." *Proc. 27th Annual Conference of the International Group for Lean Construction (IGLC 27)*, Dublin, Ireland.

Tzortzopoulos, P., Chan, P., Cooper, R., and Kagioglou, M. (2005). “Requirements management in the design of primary healthcare facilities.” *Iv Sibragec, I Elagec*, 44(1), 386-397.