Towards blockchains for architectural design: consensus mechanisms for collaboration in BIM.

DOUNAS, T., LOMBARDI, D. and JABI, W.

2019

Towards Blockchains for architectural design

Consensus mechanisms for collaboration in BIM

Theodoros Dounas[1], Davide Lombardi[2], Wassim Jabi[3]
[1] Robert Gordon University [2] Xi'an Jiaotong-Liverpool Liverpool University [3] Cardiff University
[1] t.dounas@rgu.ac.uk [2] davide.lombardi@xjtlu.edu.cn [3] JabiW@cardiff.ac.uk

We present a Blockchain collaboration mechanism on optimisation problems between distributed participants who work with building information modelling tools. The blockchain mechanism is capable of executing smart contracts, acting as a reward mechanism of independent designers attempting to collaborate or compete on optimising a design performance problem. Earlier work has described the potential integration through different levels of Computer Aided Design and Blockchain. We present an expanded version of that integration and we showcase how a team can collaboratively and competitively work, using BIM tools, through the blockchain. The original contribution of the paper is the use of the design optimisation performance as a consensus mechanism for block writing in blockchains. To accomplish that we introduce mechanisms for BIM to Blockchain Integration but also describe a special category of blockchains for architectural design and the built environment. The paper concludes with an analysis of the relationship between trust and values as encapsulated in the blockchain and how these could affect the design collaboration.

Keywords: *Blockchain, BIM, agent, collaboration, competition*

INTRODUCTION.

The paper investigates collaboration and competition in architectural design, using a model where agents are competing to create the optimum design for a given design problem, where each optimum found in the design space is used to synchronise all other agents to that solution before the computation begins anew. In the first section of the paper, we discuss issues of collaboration in BIM, trust, versioning, the notion of truth and finally performance in Building Information Modeling. In the next section, we examine the Ethereum (ETH) Blockchain as a generic Turing machine, where smart contracts and transactions change its collective decentralized state. We also discuss previous work we have conducted in CAD-blockchain integration, the concept of Decentralised Autonomous Organisations (DAO) in the blockchain and how they can be used in a design process. We then develop the case for competitive collaboration. In relation to the Ethereum

(ETH)blockchain, we discuss the notion of truth and the mechanism for consensus in the ETH blockchain, and the mechanism with which the Ethereum Virtual Machine (EVM) reaches consensus. We then argue for the application of competitive collaboration in BIM processes through smart contracts running on the EVM. We present the design optimization problem we use as a case study, the Interplanetary filesystem and how our prototype uses the blockchain to reach consensus among competitive and collaborative agents. The paper then ends with a discussion on our findings, limitations, and future work.

COLLABORATION IN BUILDING INFORMATION MODELLING

Building information modelling tools are digital tools that manage a virtual representation of a physical facility, from inception to operations. They are used by the full gamut of the AEC industry and are at the moment the prevalent mode of collaboration. One characteristic that BIM tools frequently rely on is that of trusted collaboration, as every agent in the AEC team, from the client to the architect to the consulting engineers has access to the central BIM model. However, BIM has not solved issues of trust, reliability, transparency completely. According to Brazier, Guerrero] claims of a single authoritative version of a BIM database in a project go only as far as the trust that stakeholders have in the keeper/owner of the database and the infrastructure that the database runs on. This also raises transparency and ownership issues with 'cloud' computing providers and the ownership and control of the BIM files, along with issues of version control and regression in distributed teams.

Ma et al. have found that in architectural design projects that use an integrated project delivery collaborative tasks, such as planning, work creation and execution can be automated via collaborative digital platforms. Chen et al have proposed a system for BIM collaboration via the internet, both via server client and peer to peer collaboration. Within that system, a number of teams are working in parallel in their own BIM silo, with information coordinated amongst the models in key moments in time. Kim et al have also identified time as a critical component of making decisions in BIM supported systems within the masterplanning discipline, where time-dependent metrics and multiple variants can inform the masterplan in a fragmented manner, constraining and making decision making difficult. To counter this Kim et al develop a centralized decision support system for masterplanning that collected all needed digital information into one platform that also incorporated the needed reasoning capacity for decisions. Ma et al(2), further have developed a web-based system, for construction quality inspection that also integrates BIM and a wi-fi based positioning system. One of the features highlighted in this system by stakeholders is the preservation of data in QA processes, following strict protocols, and a record of responsibilities of stakeholders that the system provides. Fittingly Hattab et al analyse information and measure data flows in BIM processes using agent-based modelling and social network analysis. Within their study, they identify fundamental conditions to be met for BIM collaboration to be successful, one of which is setting up BIM as a design "process" rather than a design "tool", where bottlenecks in the collaboration can be identified.

Value then is not only added by identifying bottlenecks but is also added in BIM design collaboration By identifying with the pattern of adding value, the member of the design team responsible, the time this takes place along with the part of the project where this takes place. In a similar study, Hattabe et al (2) identify the pattern of error creation along with the potential benefits of lean management practice in BIM processes, which are the reduction of errors and constraining their diffusion in the team.

Diraby et al have used IFC classes as a vehicle for connecting a BIM model with an energy analysis system demonstrating the analytical benefits of expanding BIMs representational capacity with performance analytics. The same objective has also been accomplished by Jabi et al, but in a much more elegant and succinct representational manner, by using non-manifold topology within a BIM system. Value proposition of tool interoperability in BIM is more widely examined by Grilo et al, where they conclude that value creation is not just tool specific in BIM but includes culture, context, values and business practices. Grilo et al also conclude that contractual issues in BIM interoperability and all the aforementioned issues are only partly addressed by current BIM practices, and only in homogeneous BIM environments.

Brazier et al have examined trust in distributed agent-based design problems, and have framed the difficulty of assigning trust in a design agent automatically. They have also detailed the extend of the problem before trust can be used explicitly in an agent-based design environment. Williams et al have examined behavior modeling and its impact on human to human interactions in collaborations, via the accumulation of data and its subsequent analysis. Guerriero et al have concluded that there exists an issue for trust within AEC virtual teams, where a higher effort and reflexivity is needed by team members to trust data that gets entered by others in the BIM system. Trust and the connection of trust with performance is also an issue which has been investigated by Brazzier in distributed design, as agents in a distributed design environment need to know which action to take in response to stimuli from other agents. Team Reflexivity, trust and collective decision making has also been analysed by Dounas et Lombardi where the distributed nature of a shape grammar and the existence of a decentralized autonomous organization operating on the Ethereum Blockchain is used to allow a distributed team to make design decisions.

From all these we conclude that BIM collaborations would benefit from a mechanism that would allow for data flows to be recorded, responsibility to be assigned to all stakeholders according to actions, data entry is to be trusted for each stakeholder, transparent tool interoperability is desired, digital information is orchestrated from diverse fragmented sources with enough trust, and the orchestration of protocols can be recorded in a trusted manner.

BLOCKCHAIN: BACKGROUND AND PRIOR WORK

Distributed ledger technologies and specifically Blockchain are essentially distributed databases between various computing nodes. Due to their distributed nature, there exists a difficulty in establishing a common, agreed, version of the truth between nodes, as one could write two different strings for example in two different computing nodes hosting each part of the database. Blockchains use a unique mechanism to establish consensus regarding which operation & transaction on a distributed database network is true and which one is not. Compared with a central database, where one queries the database directly, a Blockchain is distributed in various nodes over the network. The main consensus tools with which nodes establish the truth among them are proof of work and proof of stake. Both of these consensus tools create a new block on the Blockchain, validating one operation. This new block becomes the latest block in the chain, hence the term blockchain. Nodes participating in a blockchain network are either full nodes - those containing a full copy of the blockchain, miners - full nodes that also participate in the mining proof-of-work contest between nodes, or light clients - those that synchronise only part of the blockchain to save resources. We have chosen to implement our solutions on the Ethereum blockchain as it provides the following benefits compared to other blockchains: It behaves as a state machine, i.e. a Turing machine that allows nodes to change its state. Thus it is possible to record a variety of information on the Ethereum Blockchain. It also has the benefit of being programmable through code, either in its native language solidity or even python. Code executed on the Ethereum blockchain is called a 'smart contract' as its immutable nature equates the concept of code execution with Law.

Within the Ethereum blockchain, the consensus establishing algorithm is called Ethash, a variation on the Dagger-Hasimoto algorithm. Ethash is currently a proof-of-work algorithm and has three distinctive characteristics: ASIC-resistance, light client verifiability and full chain storage. ASIC resistance means that one does not need special computing equipment to participate in solving the computational problems that consist of the proof-of-work algorithm. In proof-of-work algorithms participating computers must produce a binary blob called a nonce, which when hashed cryptographicaly must produce a value lower than a pre-specified target threshold. Ethash begins with the preprocessed Header, which is derived from the previous block in the chain, and the current nonce. These are combined using a SHA-3 algorithm [Antonopoulos & Wood 2018] to create an initial 128byte mix. A large dataset, held in memory by all full nodes is generated every 30,000 blocks. Slices from that dataset are selected by each computing node, hashed together and then combined with the 128byte mix to generate the nonce that will be compared with the pre-selected target. The node that achieves this feature first receives a coin which is 'mined' for the purpose, hence the term 'mining cryptocoins'. The winning nonce is included in the block that will be the next block in the chain. The existence of the proof-of-work mechanism as a sufficiently hard but achievable hurdle ensures the continuing participation of mining nodes so that the transactions recorded in the blockchain are verifiable each time.

Initial blockchain systems were created so that digital distributed currencies could become possible. However, blockchains allow the computation of much more than simple additions. The full Ethereum network for example with all participating nodes (mining, full nodes, and light clients) is essentially a fully configured Turing machine. Due to the Ethereum blockchain being equivalent to a Turing machine, it has all features of Turing completeness. The positive aspect of Turing completeness is the fact that one can treat the Ethereum Blockchain as a generic computing infrastructure, capable of emulating other computers. The disadvantage would be that the Ethereum blockchain is also susceptible to the problem of incomputability, i.e not knowing for all classes of problems whether a computation will terminate with a solution or indefinitely loop with no halting mechanism. Subsequently, the Ethereum blockchain has a halting mechanism built in, in the form of computation or transaction fees. Any smart contract or code execution - transaction on the EVM needs to pay a transaction fee to be executed. If the Code causes the EVM to go into an endless loop, code execution stops as soon as the fees paid for that purpose are depleted.

In our previous work, we have discussed how one agent can establish a level of trust through combining a CAD system with smart contracts on a blockchain, and the possible levels of integration between CAD systems and a blockchain [Dounas Lombardi 2018]. We have also analysed how one can form decentralised autonomous organisations by exploiting the use of smart contracts for that purpose, and use the DAOs as a tool for design [Dounas Lombardi 2019]. Although DAOs can be currently used as good models for design governance, they are still far from being suitable for iterative exploration of issues of engineering or design performance. Due to their distributed nature, they are good vehicles for establishing collaboration within large groups, but not efficient vehicles for optimization problems. However, Jabi-Aish have created reference models for BIM, using non-manifold topology, where complex information can be represented and manipulated by a proxy model using non-manifold topology, thus greatly simplifying various issues of trust in geometric representation, transformations, and complex operations of energy analysis and structural optimization.

Design optimization performance as a consensus mechanism in blockchain type distributed ledgers.

We present a BIM to Ethereum prototype that uses design performance as a consensus mechanism to navigate issues of trust, transparency, responsibility and value creation between design agents in a distributed design environment. We have structured the prototype in a manner that allows interoperability between digital tools that can communicate with the blockchain Ethereum.

For the sake of explaining the full functionality of the prototype, we present three design agents that are working each within their own software platform. For example, for design optimization problems the agents are working with Revit/Dynamo, Rhinoceros/-Grasshopper, or Blender/Sverchok, all 3d modeling platforms with visual data flow programming capabilities. The objective is to find optimized values for a particular part of the design, say for example, structural performance. [Figure 1]

As such, in our scenario, all three agents are attempting to solve a Structural design problem that has been deployed as a smart contract at a specific address on the blockchain. The stakeholder who sets the design problem has the following options at hand: update the problem, add or withdraw a monetary balance that will go towards rewards, submit the problem, and approve or reject the solution submitted by one of the agents. All these options are structured within the smart contract on the ETH blockchain. Since saving large amounts of data on the ETH blockchain can be particularly expensive, we need another immutable, decentralised manner in which we distribute files between participants. To do so we use Interplanetary File system, IPFS, a distributed filesystem that distributes storage amongst participants. When the stakeholder sets the problem via the smart contract, she also uploads a model file on the Interplanetary filesystem and associates a cryptographic hash with the problem on the smart contract. [Figure 2]

The Structural design optimization problem the agents are trying to execute is described in the following: The process is based on the Finite Element Analysis of a simple structure adopting Karamba as FEM within the Grasshopper environment.The algorithm creates first an orthogonal grid by equally dividing a surface by a variable number of steps. Lines are then transformed into beams while the nodes of the grid become the points where to positioning the supports. For the purpose of this research we simulate the presence of a high number of designer involved in finding the optimal position of the supports by introducing Galapagos, genetic solver of Grasshopper, A random component is applied to randomize the position of the supports and connected to the solver to generate and analyse multiple solutions. The algorithm workflow then follows the standard path to run a finite element analysis by adding a simple gravity load. Cross section and material have not been specified and left as the default one by the plugin. The value used for the optimization are the displacement in the structure ob tained at the end of the analysis. Each value retrieved at the end of each loop is sent to the BC and stored as a solution provided by a potential designer participating to either a collaborative or a competitive project. [Figure 3]

The same takes place with all files in our prototype as storing information on the blockchain is particularly expensive, hence the best strategy is to only store on the blockchain, hashes of specific files, rather than the file itself, which would be very expensive computationally.

The smart contract parameter structure is simple: It includes an address in the blockchain, for example 0xEBE7e47e89129382D0837F067Ed51D318c891307 that holds the smart contract and the funds. It also includes a struct with the following parameters:

- The id of the problem;

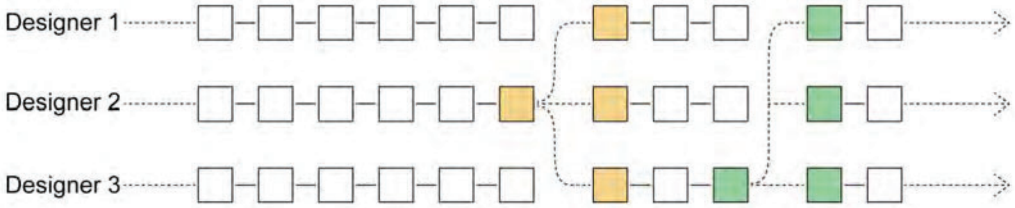

Figure 1
Three agents solving a design problem.

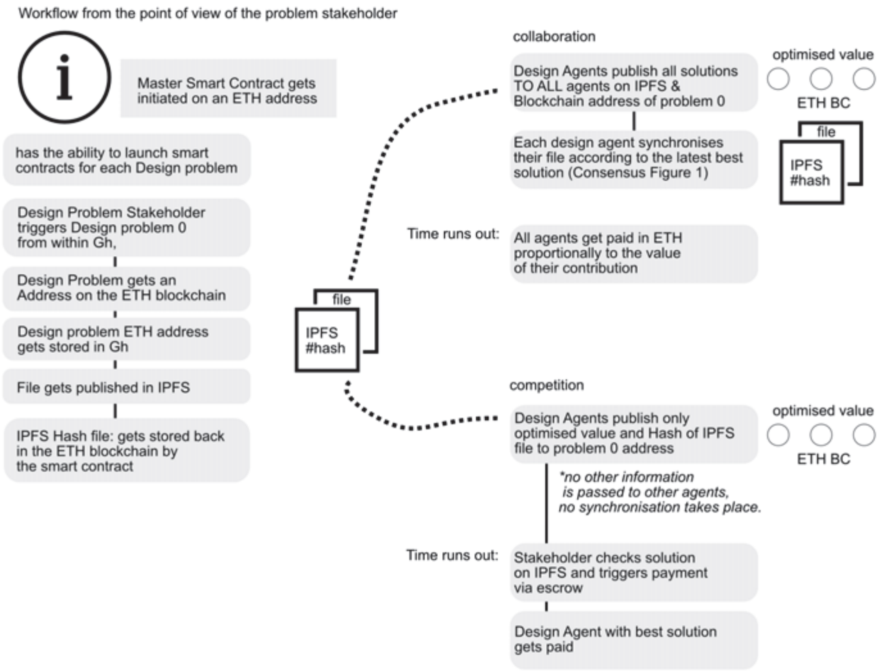
Workflow from the point of view of the problem stakeholder
i
Master Smart Contract gets initiated on an ETH address
has the ability to launch smart contracts for each Design problem
Design Problem Stakeholder triggers Design problem 0 from within Gh,
Design Problem gets an Address on the ETH blockchain
Design problem ETH address gets stored in Gh
File gets published in IPFS
IPFS Hash file: gets stored back in the ETH blockchain by the smart contract
file
IPFS #hash
collaboration
Design Agents publish all solutions TO ALL agents on IPFS & Blockchain address of problem 0
Each design agent synchronises their file according to the latest best solution (Consensus Figure 1)
optimised value
ETH BC
file
IPFS #hash
Time runs out:
All agents get paid in ETH proportionally to the value of their contribution
competition
Design Agents publish only optimised value and Hash of IPFS file to problem 0 address
optimised value
ETH BC
*no other information is passed to other agents, no synchronisation takes place.
Time runs out:
Stakeholder checks solution on IPFS and triggers payment via escrow
Design Agent with best solution gets paid

Figure 2
Design problem stakeholder data flow on the Ethereum Blockchain,

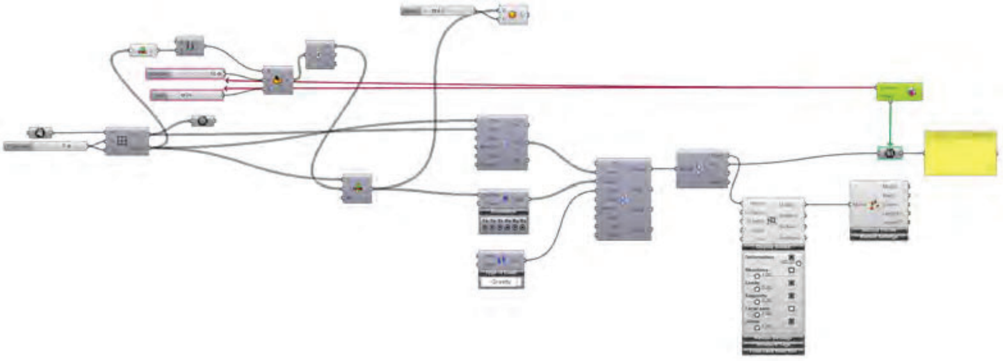

Figure 3
Structural Optimisation in Grasshopper

- The cryptographic hash of the 'problem' file on IPFS;
- The desirable inputs/outputs;
- The minimal required performance;
- The reward amount and an expiry date, after which the smart contract stops receiving input for resolution.

On the agent side, the smart contract requires three values to be provided: the user ETH address, a username and a payout ETH address, as one might want to collect monetary rewards in addresses other than the one that identifies the user. When the user goes through solving the optimization problem, a plugin on the BIM platform the agent is using submits to the smart contract the following data: the user address, the problem id, a timestamp that is used to regulate who has submitted first a solution, and a cryptographic hash that corresponds to the file the agent wants to submit as a solution.

In each of the agent's platforms, the software computes the script and reports to another node which reports to a smart contract on an 'Ethereum' blockchain via node.js, a javascript library. The smart contracts compare the value of the node either with the current optimized value and a predetermined threshold, and if the result is better than both, executes a function on the smart contract: the contract asks the agent which has found the higher value to send over her script definition. The definition, once saved in IPFS, is given a hash and is sent over to the other agents which adopt it as the current best performing script. The cycle can potentially continue indefinitely with the smart contract on the blockchain synchronizing all the agents to the best performing script of all.

Within the smart contract one can also execute payment of monetary funds to the node that has found the current optimized value. Thus there is an incentive to the distributed agents to work to find the optimized value and continue to do so. Note that when the node sends the optimization value to the blockchain, if the value is not higher than the current optimum, then the blockchain records the attempt thus creating an immutable log of all design activity. At set moments in time, all agents synchronise with IPFS and their solutions are hashed .i.e. encoded using a SHA256 algorithm, so that even the failed solutions can be recorded and their existence verified and retrieved. The smart contract terminates the problem when time runs out, or when an agent finds the best solution possible.

DISCUSSION

We believe that our work shows promise in solving issues of trust in collaboration with BIM tools and enables new collaborative and competitive modes of practice in architectural design. Our solution is able to record all design attempts, including ones that are "failed", and all positive steps towards optimisation. It also is able to show who created value when, i.e. who had a creative moment that contributed positively to solving the design problem. Within our implementation we currently show one type of software, however agents could work with different platforms, thus bringing a variety of solutions to the collaboration. This poses thought the interoperability question: when one agent succeeds in improving the required design value, their file is uploaded and saved in IPFS, which then gets transmitted to other agents to use for their own basis. If all other agents are using different software than the winning one, then a translation mechanism is needed to be able to translate a parametric script into another. Mechanisms such as topologic that can encapsulate and describe the logic underlying a BIM model are extremely valuable for this needed translation, but we have not yet tested the interoperability between BIM/visual scripting platforms. Early testing shows however promise. Our solution introduces the dimensions within which Blockchains in architectural design can operate through digital tools.

TRUST, IMMUTABILITY, VALUE CREATION.

Even though Li et al have identified BIM and Blockchain as a low maturity field in blockchain applications in the built environment, our works shows that the possibility to build such tools is valid and real. Low maturity in BIM to blockhain integration, we feel, is due to researchers not engaging with the rich digital tool making potential of modern BIM tools and the Ethereum blockchain, but rather restricting themselves to theoretical investigations. Our processes show how blockchain can be a valuable, workable infrastructure for BIM operations, where trust, immutability and legibility of design responsibilities, and much more importantly value creation, are key for the successful adoption of the technology by architectural designers. The core advantage of using blockchains in architectural design we believe will be manifest when we are able to record on the blockchain continuous loops of optimisations, for example on structure, then on energy performance, then on material optimization, plus any other design action.

REFERENCES

Antonopoulos, A and Wood, G 2018, *Mastering Ethereum*, O Reilly Media Inc, Sebastopol, ISBN 978-1-491-97194-9

Brazier, F and Wijngaards, NJ 2002 'the Role of Trust in automated distributed design', *Agents in Design 2002Key Centre of Design Computing and Cognition*, University of Sydney, pp. 71-83

Chen, HM and Hou, CC 2014, 'Asynchronous online collaboration in BIM generation using hybrid client-server and P2P network', *Automation in Construction*, 45, pp. 72-85

Cheng, X, Lu, Y, Li, Y and Xiao, J 2019, 'Quantifying and visualizing value exchanges in building information modeling (BIM) projects', *Automation in Construction*, 99, pp. 91-108

Dounas, T and Lombardi, D 2018 'A CAD-blockchain Integration strategy for distributed validated design', *eCAADe 2018*, Lodz, Poland

Dounas, T and Lombardi, D 2019 'Blockchain Grammars', *Intelligent and Informed, CAADRIA2018*, Wellington, New Zealand

El-Diraby, T, Krijnen, T and Papagelis, M 2017, 'BIM-based collaborative design and socio-technical analytics of green buildings', *Automation in Construction*, 82, pp. 59-74

Gassel, F.J.M, Láscaris-Comneno, T and Maas, G.J 2014, 'The conditions for successful automated collaboration in construction', *Automation in Construction*, 39, pp. 85-92

Grilo, A and Jardim-Goncalves, R 2010, 'Value proposition on interoperability of BIM and collaborative working environments', *Automation in construction*, 19(5), pp. 522-530

Guerriero, A and Guillame, G 2014 'Trust within AEC virtual teams', *eCAADe 2014*, Northumbria, UK

Hattab, MA and Hamzeh, F 2015, 'Using social network theory and simulation to compare traditional versus BIM–lean practice for design error management', *Automation in Construction*, 52, pp. 59-69

Hattab, MA and Hamzeh, F 2018, 'Simulating the dynamics of social agents and information flows in BIM-based design', *Automation in Construction*, 92, pp. 1-22

Jabi, W, Aish, R, Lannon, S, Chatzivasileiadi, A and Wardhana, N 2018 'Topologic: enhancing the representation of space in 3D modelling environments through non-manifold topology', *Presented at: 36th annual Education and research in Computer Aided Architectural Design in Europe (eCAADe) 2018,*, Lodz, Poland, pp. 17-21

Jabi, W, Sue, S, Theobald, P, Aish, R and Lannon, S.C 2017, 'Enhancing parametric design through non-manifold topology', *Design Studies*, 52, pp. 96-114

Kim, JI, Kim, J, Fischer, M and Orr, R 2015, 'BIM-based decision-support method for master planning of sustainable large-scale developments', *Automation in Construction*, 58, pp. 95-108

Li, J, Greenwood, D and Kassem, M 2019, 'Blockchain in the built environment and construction industry: A systematic review, conceptual models and practical use cases', *Automation in Construction*, Vol. 102, 06.2019, pp. 288-307

Ma, Z, Cai, S, Mao, N, Yang, Q, Feng, J and Wang, P 2018, 'Construction quality management based on a collaborative system using BIM and indoor positioning', *Automation in Construction*, 92, pp. 35-45

Ma, Z, Zhang, D and Li, J 2018, 'A dedicated collaboration platform for Integrated Project Delivery', *Automation in Construction*, 86, pp. 199-209

Preisinger, C and Heimrath, M 2014, 'Karamba—A Toolkit for Parametric Structural Design', *Structural Engineering International*, 24(10.2749/101686614X13830790993483)

Williams, M, Burry, J and Rao, A 2015 'Understanding Face to face interactions in a collaborative setting, methods and applications', *CAAD Futures 2015*, Sao Paolo, Brazil

[1] https://github.com/ethereum/wiki/wiki/Dagger-Hashimoto

[2] https://github.com/ethereum/wiki/wiki/Ethash