A Block-chain Oriented Model Driven Framework for handling Inconsistent Requirements in Global Software Development

Nayab Gull*
Department of Computer & Software Engineering, College of E&ME,
National University of Sciences and Technology (NUST), H-12, Islamabad,
Pakistan

Muhammad Rashid
Computer Engineering Department,
College of Computer and Information Systems, Umm Al-Qura University,
Makkah, Saudi Arabia

Farooque Azam
Department of Computer & Software Engineering, College of E&ME,
National University of Sciences and Technology (NUST), H-12, Islamabad,
Pakistan

Yawar Rasheed
Department of Computer & Software Engineering, College of E&ME,
National University of Sciences and Technology (NUST), H-12, Islamabad,
Pakistan

Muhammad Waseem Anwar
Department of Computer & Software Engineering, College of E&ME,
National University of Sciences and Technology (NUST), H-12, Islamabad,
Pakistan

ABSTRACT

Requirements engineering (RE) is one of the most important phases in software development process. Therefore, the handling of inconsistent/ conflicting requirements is considered as a major issue in the requirements phase. Due to the evolving practices of Global Software Development (GSD), where the development team is geographically spread, the phase of RE is more prone to inconsistencies. Many approaches have been suggested to overcome the problem, however, these solutions are very limited in their scope and does not fit the peculiarities of GSD configurations. On the other hand, Block Chain methodology has shown promising results in various domains and has very sophisticated features like transparency and decentralization to support the management of inconsistent requirements. Similarly, Model Driven Software Engineering (MDSE) is marked with abstraction and reducing complexity which may be benefited to handle inconsistent requirements. Consequently, this article introduces a Block chain Oriented Model Driven (BOMO) framework by integrating the concepts of RE and Block Chains in the context of MDSE. This allows the effective management of inconsistent requirements through block chain technique with simplicity as offered by MDSE. As part of the framework, a metamodel is proposed which has been subsequently evolved into a Sirius graphical modelling tool. The applicability of the proposed framework has been demonstrated via a case study. Experimental results prove that the proposed framework may be used with sufficient reliability and can be further evolved to handle inconsistent requirements in a promising manner.

CCS CONCEPTS

• **Software and its engineering**; • **Software organization and properties**; • **Software system structures**; • **Software system models**; • **Model-driven software engineering**;

KEYWORDS

Model Driven Architecture (MDA), Meta-modelling, Block Chains, Inconsistent Requirements, Global Software Development

ACM Reference Format:
Nayab Gull, Muhammad Rashid, Farooque Azam, Yawar Rasheed, and Muhammad Waseem Anwar. 2021. A Block-chain Oriented Model Driven Framework for handling Inconsistent Requirements in Global Software Development. In *2021 10th International Conference on Software and Computer Applications (ICSCA 2021), February 23–26, 2021, Kuala Lumpur, Malaysia.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3457784.3457799

1 INTRODUCTION

Requirements are usually handwritten and are subjected to various issues and bugs. If a bug remains unhandled, then it impacts negatively on the success of the final product. This issue seems more in Global Software Development (GSD). GSD is a kind of Distributed Software Development (DSD) in which the software teams are distributed beyond the limits of a Geographical bound. Consequently, GSD setup face harder challenges [1] as compared to local software development (LSD). Second and major factor for inconsistency is the communication gap between the dispersed stakeholders.

Although many techniques for the removal of inconsistent requirements have been proposed which help to manage inconsistency issues well for local software development but do not address the needs of GSD. Nicolas et al. [2] conducted a systematic literature review to identify the risks in requirement engineering phase in GSD. Mezghani et al. [3] has used k-means clustering algorithm using the software SEMIOS to detect the inconsistent requirement. Certain authors also suggested the model driven approach for automatic detection and tracking of inconsistencies in the software requirements. But requirements inconsistency in GSD is still a big challenge.

[*] ngull.cse19ceme@ce.ceme.edu.pk

ICSCA 2021, February 23–26, 2021, Kuala Lumpur, Malaysia
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8882-5/21/02…$15.00
https://doi.org/10.1145/3457784.3457799

By considering the GSD configuration, blockchain technology that is assumed to control the world very soon can be observed as a solution for removing requirements inconsistencies in GSD. This technology possesses a variety of robust features, that's why almost every domain in the world is attracting towards it to solve the inherent problems. Decentralization, faster settlement and transparent nature of this technology has the potential to solve the issues which GSD had been facing since decades.

The evolving concepts of Model Driven Software Engineering provides better abstraction and Modularity [4]. It is reducing the complexity by enhancing understanding of the complex system among stakeholders [5]. Therefore, in this paper, by exploiting the advantages of model driven approach and famous blockchain technology, a novel framework named Blockchain Oriented Model (BOMO) is proposed. It is a solution to the challenges faced by the GSD. Proposed framework permits the dispersed team of GSD to connect to each other in such a way that the effort of each team member is visible to the whole network. A voting consensus algorithm is used to select and finalize the specification of requirements. Therefore, it specifies only the bug free requirements and ensures the high standards of quality in RE.

Proposed framework consists of a metamodel which is an M2 level Ecore model defined using Eclipse Modelling Framework (EMF) [6], [7]. As tool support, a graphical modelling tool has also been developed using Sirius. This tool provides a full abstraction and reduces the implementation complexity of the system. Graphical modelling tool also consists of drag drop palette. The customized nodes and edges that are created for the framework can also be dragged and dropped in order to model various scenarios. The proposed framework is validated through a benchmark case study.

2 LITERATURE REVIEW

Many researchers provide the effective techniques towards the issue of inconsistency. Moitra et al. [8] introduced a tool called ASSERT (Analysis of Semantic Specifications and Efficient generation of Requirements-based Tests) for capturing requirements, backed by a formal requirements analysis engine. Riaz et al. [9] surveyed on the currently available tools and techniques for the automatic detection of natural language ambiguities from software requirements. Demuth et al. [10] is using cloud-based approach in order to check inconsistency in the requirements. Yang et al. [11] has used RM2PT tool to automatically generate prototype through requirement model and confirms the requirement validation.

Only few researchers have found who show interest towards providing solution for distributing working mode (GSD). Mighetti et al. [12] proposed a methodology using a lexicon and scenarios models in order to mitigate the inconsistency threats in the requirements in GSD environment. Hossain et al. [13] analysed the scrum practices in order to solve the issues in requirement phase in context of GSD. Anukula et al. [14] manages the inconsistency in GSD by using the cost estimation model.

Model is the graphical representation and description of a system. Metamodel (i.e. domain model) describes the main concept of a certain domain [15], [16], [17]. It helps to reduce the complexity of a system by defining things at an abstract level and provides the understanding of complex systems [18]. The work in [19] provides a model driven approach in the domain of software requirement engineering in order to specify and trace the software requirements. Kanth et al. [20] proposed a model driven framework for automatic detection of inconsistencies in the requirement phase.

It is analysed that there exist several approaches regarding GSD inconsistent requirements, but they do not provide the exact solution for the GSD configuration. Furthermore, MDA also employed in GSD inconsistent requirements, but it can help only the local software development team. As, it is essential for the GSD team to involve stakeholders as much as possible and to have clear communication structure between them. Therefore, literature does not give such a mechanism to achieve exact solution.

On the other hand, blockchain, a decentralized network serving a lot to the world due to its strong properties. All the transactions occur in this technology are visible to all the participant nodes. A new block is added to the chain on the basis of consensus algorithm which is used in a particular chain. Up till now many algorithms have been proposed. It's up to the developer to choose suitable consensus algorithm for the application to be developed. Many authors have proposed a bundle of applications by using this technology i.e. to overcome personal data leakage [21], to ensure the trustworthiness of health care system [22], to ensure the quality of food industry [23], to preserve the data for AI developers [24].

Therefore, blockchain features are highly supportive to address the problem of GSD inconsistent requirements. However, it may add additional complexities that why it is applied so far. In this situation, an effective solution can be developed for GSD inconsistence requirements by combining the concepts of MDA and block chains systematically. Therefore, in this article by considering the advantages of model driven approach and attractive features of blockchain, both concepts are utilized to develop a complete framework that serves the GSD to cope up with the inconsistency in the requirements.

3 PROPOSED FRAMEWORK

Proposed framework BOMO gracefully handles the issue of inconsistent requirements for the Global Software projects. This section provides the detail of the framework.

3.1 High Level View of Proposed Framework

1 shows the high-level view of the proposed framework 'BOMO'. In this framework, all the nodes (participants) associated in a particular project are connected in a peer-to-peer network of block-chain. There is no discontinuity anywhere, as each individual node is part of the entire block chain application.

In proposed BOMO framework, there are two types of major nodes. 1) **Stakeholders** are the development team members which invoke the transactions i.e. uploading the requirements and can validate the transactions of others, 2) **Block-chain Manager (BCM)**, responsible to add a new block. All BCMs and stakeholders when get added in a block-chain network will get a private Id in order to have the identity on that network. Identity management is also done by the BCM of every cluster. Main points of this framework are summarized in the following steps and flow is provided in 2

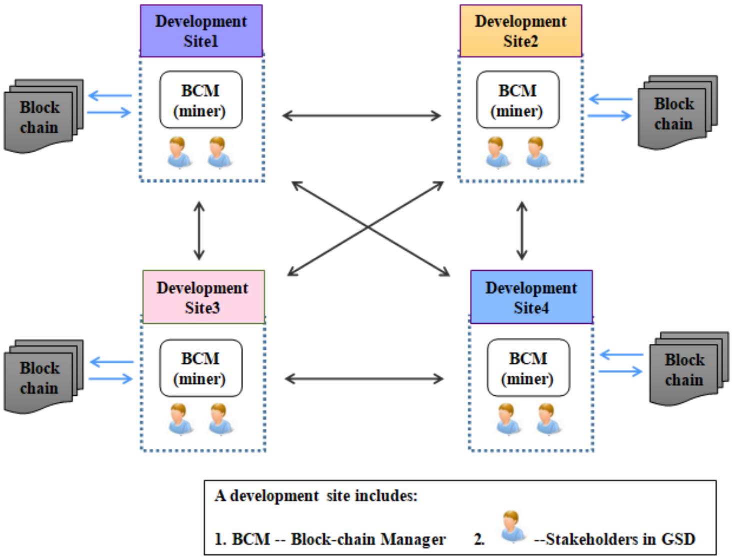

Figure 1: High Level View of BOMO

1. A stakeholder from any development site uploads the particular requirement.
2. That upcoming requirement is then added into the requirement pool.
3. Each BCM will pick the requirements from the pool and put them into the block according to the capacity of blocks. Candidates blocks are then broadcasted to the network and transaction invokes.
4. All the candidate blocks are validated by all the BCM's present on the network. As candidate blocks are fed into the consensus Voting protocol.
5. Only one block is selected which consists of bug-free and consistent requirements.
6. Status of particular requirement either accepted or rejected is then notified to the whole network in order to proceed for subsequent transactions.

More than one BCM can create a block at the same time by putting particular requirements in their candidate blocks. As every BCM will create block and adds different requirements in it. But it might be possible that a BCM is adding those requirements which are either inconsistent or may not be needed yet to add in a chain for a couple of reasons. So, if a candidate block of a BCM fails, then next BCM who is done with its candidate block will broadcast and wait for verification from all the BCMs. Therefore, this is a type of voting mechanism in which all BCMs are voting that which block is best suitable to add into a chain. Incentives given to BCM which successfully adds a block encourages all other BCMs to take part in a consensus algorithm. The BCM whose block has finally selected using POV consensus protocol creates the hash against that block. After that, this block broadcasted to every node for checking the preconditions and post-conditions of the block. When validation process is completed, then that block is added to a block-chain successfully. And stakeholders now add further requirements to create subsequent blocks.

3.2 Proposed Meta-Model

In this section, description of proposed metamodel is provided**. Figure: 3** shows the proposed BOMO metamodel. This framework is efficiently predicting the consistent and inconsistent requirements.

- **BOMO Framework**: It is the root class representing a system and composing all of the other classes in it.
- **Blockchain**: Private permissioned blockchain network over permission-less network is preferred for this framework. Light weight blockchain framework [25] is used in this proposed framework. In lightweight framework, clusters of participant nodes are formed, and some privileges are given to only one chosen participant of a cluster in order to maintain the security.
- **Site**: Every organization that belongs to a particular country or location is one cluster or development site. Stakeholders are grouped into clusters based on their demography. Each development site can have multiple stakeholders like requirement analysts, developers and the contributors nearer to that organization. There are 1,2,3, . . .N development sites in the framework and each site provides requirement collectively coming from the number of stakeholders.
- **Blockchain Manager (BCM):** Within each cluster, one node is selected as the Block-chain Manager (BCM). Experts of a proposed software project from each cluster is chosen as BCM. Some nodes i.e. some secondary stakeholders can't keep the copy of ledger because of security issues, they can query the data from their BCM's ledger copy. If every node

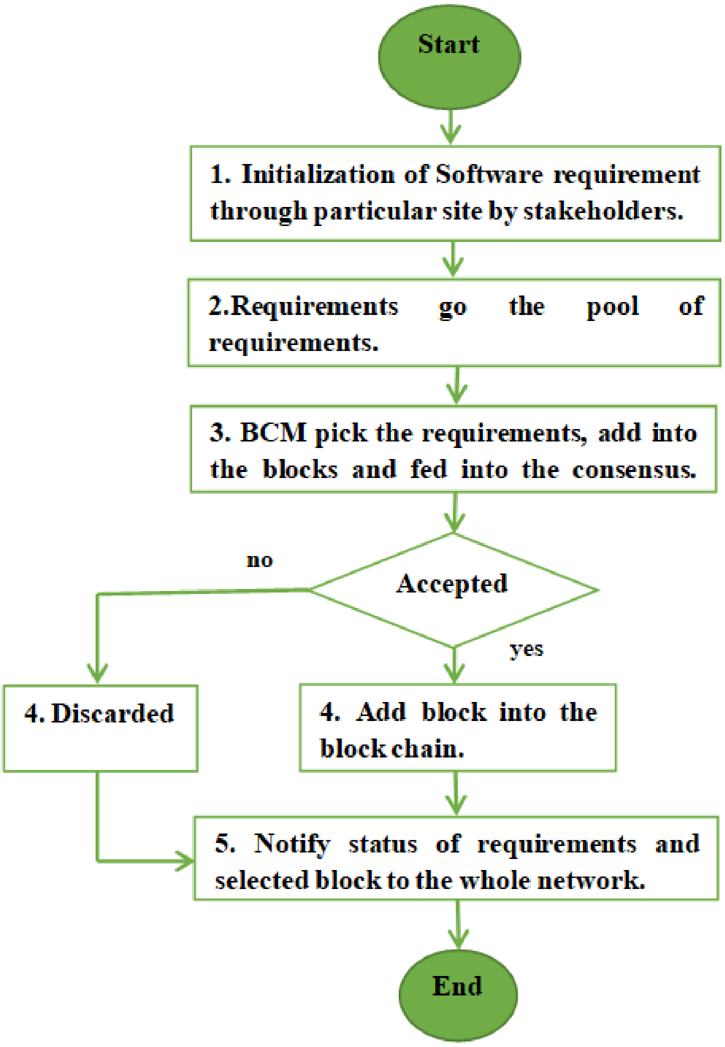
Start
1. Initialization of Software requirement through particular site by stakeholders.
2.Requirements go the pool of requirements.
3. BCM pick the requirements, add into the blocks and fed into the consensus.
Accepted
no
yes
4. Discarded
4. Add block into the block chain.
5. Notify status of requirements and selected block to the whole network.
End

Figure 2: Working Algorithm of Proposed Meta-Model

keeps the copy of ledger, then there would be a computational overhead. It depends on the nature of project, that who is permitted to keep the copy of a whole ledger in each development site.

- **Transaction**: This concept is a manifestation of a blockchain's feature. Transaction data comprises the requirements that is provided by the different stakeholders. This class is related to consensus-voting protocol class.
- **New Block**: This concept is a manifestation of a blockchain's structure. Requirements will be saved into the blocks of blockchain. Blocks includes transaction data i.e. requirements and the hash of the previous block.
- **Requirements:** This class is the representation of software requirements.
- **Stakeholders**: This concept is a manifestation of real-world stakeholder. Number of stakeholders belonging to different sites present in this network has the authority to provide the requirements suitable for the upcoming release. This class can be related to number of transactions zero or one.
- **Broadcast**: When number of requirements get together in the pool, then BCM belonging to that particular site puts them into the candidate blocks and broadcast that block over the network. Maximum of three requirements are allowed to put into the candidate block in proposed framework.
- **Consensus protocol**: It is the main concept used in blockchain technology in order to mine a block to append the chain. Consensus protocol in this framework is Proof-Of-Vote (POV), which is best suitable for the management of

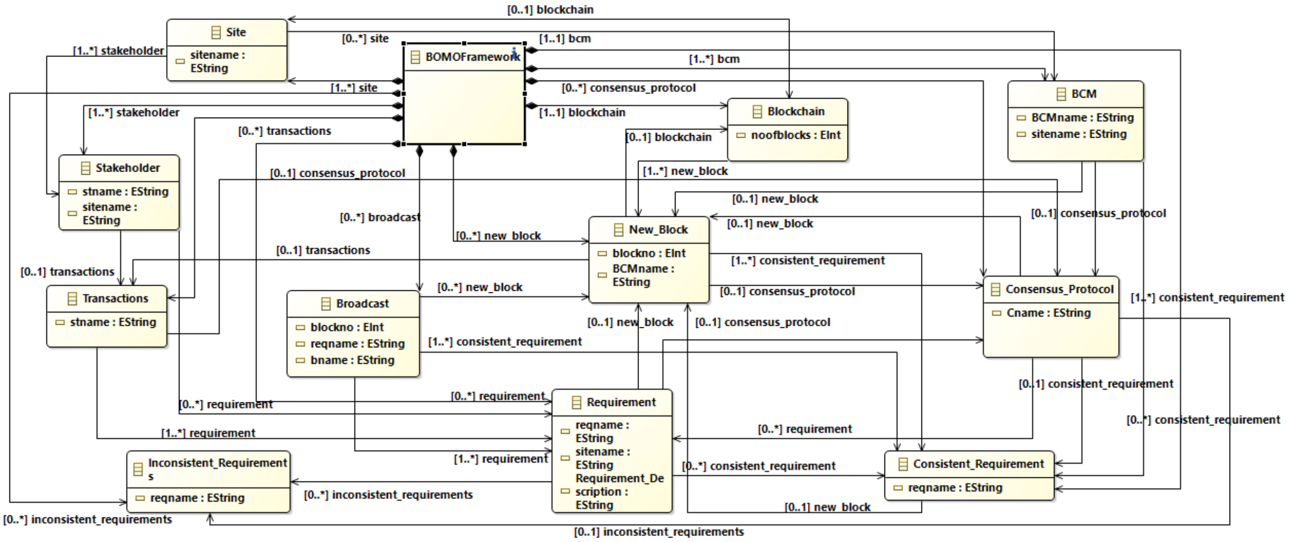

Figure 3: Proposed BOMO Meta Model

software requirements in terms of inconsistency. This class is related to the classes of consistent, inconsistent requirements and New block.

- **Consistent Requirement:** Requirements that are accepted by the consensus of all the BCM's present in the network is considered as consistent requirement. Those requirements that do not conflict to the previous stored requirements.
- **Inconsistent Requirement:** Those requirements that are declared as rejected requirements by the BCM's are inconsistent requirements.

This framework is best suitable for the removal of inconsistent requirement in the GSD configuration. Proposed framework ensures the specification of best quality software requirements. It has been stated earlier that nature of the proposed framework is flexible and can be tailored according to the nature of the projects that will be developed using this framework. Global software development industry can utilize this framework to solve many more requirements related problems.

3.3 Tool Support

As tool support, we have developed a graphical modeling tool with a user-friendly drag drop palette. This tool also has a diagram area where various concepts of the meta model can be instantiated with the help of drag drop palette. Graphical modeling tool has been developed using Sirius in Obeo Designer Community Edition. 4 shows the case study model developed using our tool.

4 CASE STUDY

A scenario of a multinational company is taken which is using the global software development configuration. The company name is company X. X is developing a product for a piping industry. The product they are going to build is AutoPIPE. It is a product which is used for pipe stress analysis. This product requires core programming developers as well as the R & D developers. R & D developers are actually mechanical engineers with basic programming knowledge. Their purpose is to provide and implement the actual industry knowledge they have and collaborate with core programmers for the accurate development of modules according to the piping industry needs. This product provides a stress analysis of pipes present in a plant, nuclear reactor or any other piping components.

There are multiple branches of company X spread over the various geographical locations in the entire world. This project stakeholders include: Requirement analysts: To gather the requirements according to the industry needs, Developers: To develop the desired product, R & D Developers: Provide industry knowledge.

These stakeholders are working collectively while sitting at diverse locations. Company is using proposed framework for requirement specification for the Auto PIPE product. In this scenario multiple stakeholders provide the requirements by collecting from clients according to the proposed software. There are four branches involved to develop this product. Every organization needs to install software to use proposed framework and create blockchain. We assume that all the setup has been created for maintaining a blockchain. Using the proposed methodology, multiple nodes upload the requirements for Auto PIPE just like shown in 1. Unlike the traditional requirement gathering and specification methods, proposed framework provides a synchronized system in which every stakeholder takes part in writing and validating the requirements. Developers are also validating and checking the requirement in terms of requirement realism, whether in reality a particular requirement can be developed or not.

A Stakeholder from a development site initiates a requirement R and broadcasts to the whole network. Every node from each site validates R by checking whether it meets project need or not. Likewise, all the stakeholders initiate their requirements shown in 4

1. BCM1 added R6 (which initiated by stakeholder1) in its candidate block.

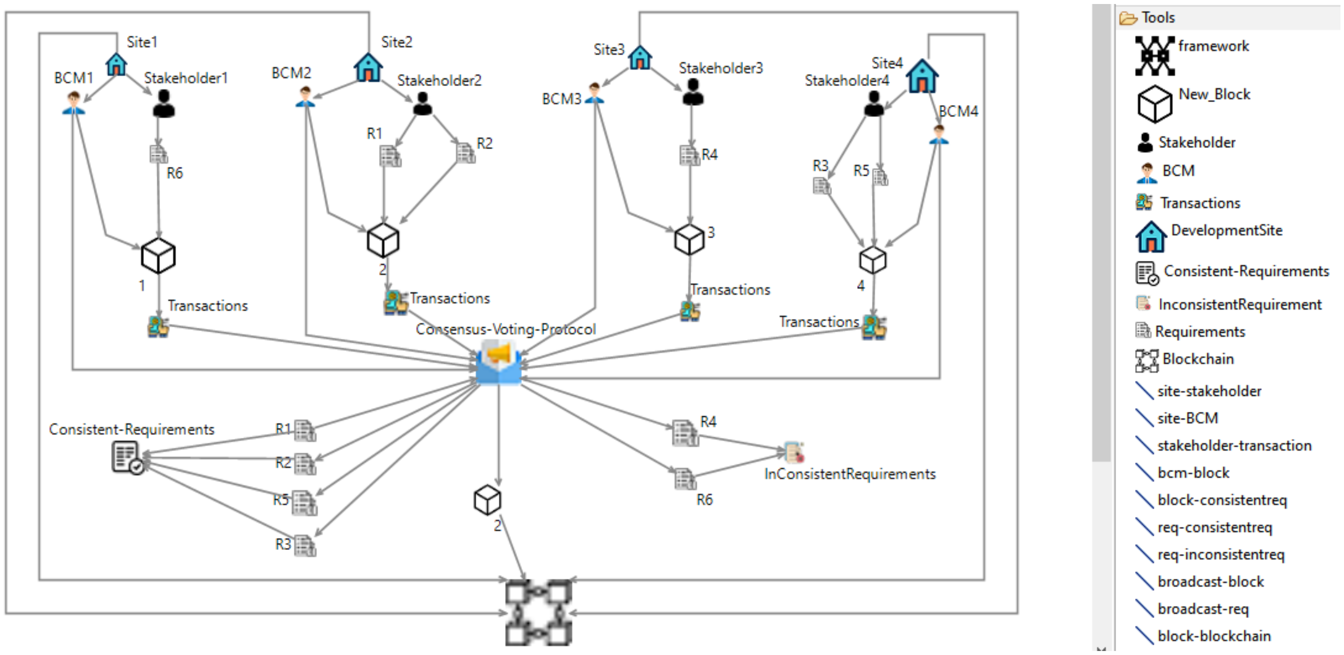

Figure 4: Case Study of Proposed Meta model

2. BCM2 added R1 and R2 (Those initiated by stakeholder2) in its candidate block.
3. BCM3 added R4 (which initiated by stakeholder3) in its candidate block.
4. BCM4 added R3 and R5 (which initiated by stakeholder4) in its candidate block.

All the four BCMs broadcasted their candidate blocks in which they have added particular requirements set shown in 4. Only one block is chosen at a time by consensus protocol. According to the case study, R1 and R2 are the two requirements which added into the block2 of site2, both these requirements are accepted, and their associated candidate block has accepted by the consensus-voting-protocol. R4 found to be incongruent with R2, so its associated block has been rejected, and it is subjected towards inconsistent requirement and discarded. R3 and R5 are accepted and their associated candidate block added to the blockchain. R6 found to be inconsistent with the R5 and gets rejected.

Accepted blocks are then broadcasted as well to the whole network in order to notify the status of requirements and blocks. Then, BCM try to make further subsequent blocks. This was a small setup which has used the proposed framework. This framework could be useful for larger setup and for large projects as well. In this case study, it may be noticed that not only inconsistent but also conflicting, redundant and irrelevant requirements can be removed by mutual consensus of the block-chain oriented Approach.

5 DISCUSSION AND LIMITATIONS

Proposed framework exploit blockchain's feature for inconsistent requirements with simplicity through MDA concepts. By applying the blockchain technology in the framework, system provides a synchronized environment in which stakeholders from diverse backgrounds can connect to each other and observe each activity in a formal platform. To reduce the complexity and in order to achieve the abstraction, model driven approach is used. It helps requirement engineer to understand the system in detail.

Results show that proposed approach significantly helps the global software industry while dealing with inconsistent requirements. Furthermore, it has been found that this effective approach not only solves the inconsistency of the requirements, but also caters to other requirement related problems i.e. unrealism and requirement prioritization to some extent. In addition to this, the consensus algorithm of blockchain is very effective mechanism and contributes a lot in helping the distributed stakeholders. Every BCM on solving the proof-of-Vote algorithm gets incentives like the public blockchain network. In this way, every organization tries to work better to add requirements in a block which must be acceptable through all the nodes. In future, we will apply formal verification of proposed approach by using voting protocol in order to further improve the management of inconsistent requirements.

6 CONCLUSION AND FUTURE WORK

This article introduces a novel blockchain oriented model driven framework i.e. BOMO framework, that can be reliably used and further evolved for the management and handling of inconsistent requirements in GSD. The applicability of the proposed framework has been demonstrated by modelling a case study scenario and relevant concepts of RE and block chains have been instantiated using our Sirius graphical modelling tool. This gives fair graphical visualization of the complex scenarios as well which is the real essence of MDSE. In future we intend to incorporate more detailed concepts pertaining to block chains in our meta model. Enhancement of the tool as well as writing Model to text transformation for transforming the modelled scenario into a platform specific code is also a milestone to be achieved.

ACKNOWLEDGMENTS

We acknowledge the support of KACST (King Abdul-Aziz City for Science and Technology) and STU (Science and Technology Unit) Makkah, Saudi Arabia (Grant Number 14-ELE1049-10)

REFERENCES

[1] N. Saleem, S. Mathrani, N. Taskin, “Understanding the Different Levels of Challenges in Global Software Development”, 14th International Conference on Global Software Engineering , Montreal Canada, 2019.

[2] J. Nicolas, J. M. C. D. Gea, B. Nicolas, J. L. FernaNdez-Aleman, “On the Risks and Safeguards for Requirements Engineering in Global Software Development, Systematic Literature Review and Quantitative Assessment”, IEEE Access, vol. 6, pp. 59628-59656, 2018.

[3] M. Mezghani, J. Kang and F. Sèdes, "Industrial Requirements Classification for Redundancy and Inconsistency Detection in SEMIOS," IEEE 26th International Requirements Engineering Conference, Banff, AB, 2018.

[4] Y. Rasheed, F. Azam, and M. W. Anwar. “A Novel Framework and Tool for Multi-Purpose Modeling of Physical Infrastructures." Proceedings of the 12th International Conference on Computer Modeling and Simulation, 2020.

[5] M. W. Anwar, M. Rashid, F. Azam, M. Kashif and W. H. Butt, “A Model-Driven Framework for Design and Verification of Embedded Systems through SystemVerilog”, Design Automation for Embedded Systems, vol. 23, no. 3-4, pp. 179-223, December 2019.

[6] M. Rashid, M. W. Anwar, A. M. Khan, “Towards the Tools Selection in Model Based System Engineering for Embedded Systems - A Systematic Literature Review”, Journal of Systems and Software, vol. 106, pp.150-163, May 2015.

[7] M. Rashid, M. W. Anwar, “A systematic investigation of tools in model-based system engineering for embedded systems”, 11th IEEE System of Systems Engineering Conference, pp. 1-6, Kongsberg, Norway, June 2016.

[8] A. Moitra, K. Siu, A. Crapo, H. Chamarthi, M. Durling, M. Li, “Towards Development of Complete and Conflict-Free Requirements”, *26th International Requirements Engineering Conference*, Banff AB Canada, 2018.

[9] M. Q. Riaz, W. H. Butt and S. Rehman, “Automatic Detection of Ambiguous Software Requirements, An Insight”, *5th International Conference on Information Management*, Cambridge United Kingdom, 2019.

[10] A. Demuth, M. Riedl- Ehrenleitner and A. Egyed, “Efficient detection of inconsistencies in a multi-developer engineering environment, *31st IEEE/ACM International Conference on Automated Software Engineering,* Singapore, 2016.

[11] Y. Yang, W. Ke and X. Li, "RM2PT: Requirements Validation through Automatic Prototyping,"*27th International Requirements Engineering Conference (RE),* Jeju Island, Korea (South), 2019.

[12] J. P. Mighetti and G. D. S. Hadad, “Requirements Engineering Process Adapted to Global Software Development, CLEI Electronic Journal, vo;. 19, no. 3, 2016.

[13] E. Hossain, P. L. Bannerman and D. R. Jeffery, “Scrum Practices in Global Software Development, A Research Framework”, *International Conference on Product Focused Software Process Improvement,* Sydney, Australia, 2011.

[14] J. M. Anukula and S. M. Perumal, “Analog-Based Cost Estimation for Managing Inconsistency” *International Journal of Research Sciences and Advanced Engineering* , vol. 2, no. 3, 2012.

[15] M. W. Anwar, M. Rashid, F. Azam, M. Kashif, “Model-Based Design Verification for Embedded Systems through SVOCL: An OCL Extension for SystemVerilog”, Design Automation for Embedded Systems, vol. 21, no. 1, February 2017.

[16] M. W. Anwar, M. Rashid, F. Azam, A. Naeem, M. Kashif, W. H. Butt, “A Unified Model Based Framework for the Simplified Execution of Static and Dynamic Assertions Based Verification”, IEEE Access, vol. 8, pp. 104407-104431, June 2020.

[17] M. W. Anwar, M. Rashid, F. Azam, A. Naeem, M. Kashif, W. H. Butt, “A Unified Model Based Framework for the Simplified Execution of Static and Dynamic Assertions Based Verification”, IEEE Access, vol. 8, pp. 104407-104431, June 2020.

[18] Y. Rasheed *et al.* "A Novel Model Driven Framework for Image Enhancement and Object Recognition." *International Conference on Information and Software Technologies*, 2020.

[19] J. Jézéquel, "Model-Driven Engineering for Software Product Lines", *International Scholarly Research Notices*, vol. 2012, Article ID 670803, 24 pages, 2012.

[20] A. A. Rao, T. V. RajiniKanth and G. Ramesh, “A Model Driven Framework for Automatic Detection and Tracking Inconsistencies”, Journal of Software, vol. 11, no. 6, 2016.

[21] N. Daveze, R. Dubaele, J. Hokayem, M. Nussbaum and F. CarvalhaisSanches, “Block-Chain-Based Personal Data Hosting”, *9th International Conference on Software Engineering and Service Science*, Beijing, China, 2018.

[22] M. S. Christo, A. M. A., P. S. G., P. C. and R. K. M., "An Efficient Data Security in Medical Report using Block Chain Technology," 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2019, pp. 0606-0610

[23] W. Yu and S. Huang, "Traceability of Food Safety Based on Block Chain and RFID Technology," 2018 11th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 2018, pp. 339-342.

[24] N. Baranwal Somy *et al.*, "Ownership Preserving AI Market Places Using Blockchain," 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 2019, pp. 156-165.

[25] L. Ismail, H. Materwala and S. Zeadally, "Lightweight Blockchain for Healthcare", *IEEE Access*, vol. 7, pp. 149935-149951, 2019.