Use of Model Based Systems Engineering To Support Bricklaying Robot In Construction Automation

Mr. Mahalingam PP[1] , Srihari P[2], Abdul Kathar Jailani M[3], Vijaya Sekar P[4]
[1]Assistant Professor, Dept of Mechatronics Engineering
[2, 3, 4]Dept of Mechatronics Engineering
[1, 2, 3, 4] Dr. Mahalingam College of Engineering and Technology, Pollachi – 642 003, India.

Abstract- *This paper presents a Definition & Decomposition phase of SysML model and it's simulation with the use of Model-Based Systems Engineering (MBSE) approach to support the design and development of a bricklaying robot system for construction automation. The use of MBSE enabled the creation of a comprehensive model of the system, which facilitated the identification of requirements, design decisions, and trade-offs. The model also provided a platform for communication and collaboration among the various stakeholders involved in the project. The paper highlights the benefits of using MBSE for the development of complex systems, and how it can contribute to the success of construction automation projects. The results of the simulation of the systems Model demonstrate the effectiveness of MBSE in supporting the design and development of bricklaying robot systems, and suggest that it can be applied to other construction automation projects.*

Keywords- SysML, Model-Based Systems Engineering, MBSE, Bricklaying Robot, Construction Automation.

I. INTRODUCTION

Construction automation has been gaining traction in recent years due to the need for increased productivity, efficiency, and safety on construction sites. One area of automation that has seen significant progress is bricklaying, with the development of bricklaying robots that can perform the task faster and with greater accuracy than human workers. However, the design and development of such systems can be complex and challenging, requiring the integration of multiple components and subsystems.

Model-Based Systems Engineering (MBSE) is a methodology that employs models to record and control system complexity across the course of the system's lifecycle. Systems engineering generally refers to a V-Shaped method with definition & decomposition phase, design & implementation phase and verification & validation phase. It gives you a way to represent, evaluate, and communicate system requirements, designs, and choices, which can produce better results and reduce errors. We propose a case study in this paper on the application of MBSE to support the modelling of definition & decomposition phase for design and development of a bricklaying robot system for construction automation.

The systems model and it's simulations of the Bricklaying Robot demonstrates how MBSE can be used to create a comprehensive model of the system, which captures all relevant information in a single place. This model enables stakeholders to analyse and simulate the system's behaviour and performance, which can help identify potential issues and optimize the design. The model also serves as a platform for communication and collaboration among the various stakeholders involved in the project, which can improve the project's overall efficiency and effectiveness.

Overall, this project highlights the benefits of integrating MBSE approach for the development of complex systems in the construction industry, and how it can contribute to the success of automation projects. The results of the project demonstrate the effectiveness of MBSE in supporting the design and development of bricklaying robot systems and suggest that it can be applied to other construction automation projects.

II. LITERATURE SURVEY

[1] Zakaria Dakhli & Zoubeir Lafhaj on Robotic mechanical design for brick-laying automation, The design criteria for a brick-laying robot included the need that it should be able to build a wall out of cinder blocks on its own. The paper outlines factors to take into account throughout the full robot implementation process, from material input and stock management to the robot assembly process. The head of the brick-laying robot was included in the design. Final comparisons were done between the robot and conventional methods for masonry assembly in terms of price and productivity. The outcomes showed that the developed robot increased production while reducing waste and optimising time and cost deliverables.

[2] G. Pritschowa, M. Dalackera, J.Kurz a and M. Gaenssleb on Technological Aspects in the Development of a Mobile Bricklaying Robot, A hypothetical situation for the on-site use of a man-machine system made up of a mobile bricklaying robot and a knowledgeable worker is presented. The application of thin-bed mortar can be done automatically, and this process is explained and experimentally proven. Additionally, a device that combines technological processes like calibration of brick position, determination of material tolerances, and application of bonding substance in a single unit is explained in detail, as is the vacuum handling system of the bricklaying robot.

[3] Hannu Lehtinen, Esko Salo, Heikki Aalto on Outlines of two masonry robot systems, On the topic of masonry robotics, functional needs, technological requirements, and economic viability have been looked into. A modular SCARA-based system and a six D.O.F. articulated configuration are the two options that are discussed. With actual partition wall sizes, market prices of $ 310 000 and $ 350 000 have been calculated for the systems, with a payback period of 6 years. Economic study reveals that the size of the partition wall owing to human preparation works, which need a relatively consistent amount of time, is a crucial element in the viability of such automatic masonry systems. The final result, a partition wall, must be constrained in order to achieve sufficient technical and financial viability.

Fig. 1 SAM100 Bricklaying Robot

[4] Andrew J. Madsen on The SAM100: Analyzing Labor Productivity, The majority of the masonry workforce on a project can be replaced by the brick-laying robot SAM100, which is shown in the Fig. 1. On masonry-intensive projects, using this method could be very advantageous. The advantages and disadvantages of the SAM100 on the Jay and Susie Gogue Performing Arts Centre project at Auburn University in Alabama will be examined in this essay. This article focuses on the qualitative information related to its integration into the project when analysing its benefits. It will shed light on the benefits and drawbacks of the robot as well as some difficulties encountered when using this piece of technology. This paper's goal is to present important details about the SAM100's use from a project that employed this tool and offer recommendations for future applications.

[5] Aleksander Buczacki on An integrative Model Based Systems Engineering (MBSE) and lean-based approach for development of new complex products, A successful New Product Development (NPD) and The implementation process is essential for each company's construction of a competitive advantage. Due to this, businesses are increasingly looking for solutions. and resources for enhancing NPD procedures. The integrated Model Based Systems Engineering (MBSE) and lean approach to NPD is the topic of this study. The use of MBSE and Lean is inclined in varying degrees during each stage of the NPD process. The results of the study indicate that there is a lack of broad adoption of sophisticated MBSE and lean tools and procedures in the creation of new goods.

[6] L. Lemazurier, V. Chapurlat, A. Grossetête on An MBSE Approach to Pass from Requirements to Functional Architecture, Systems Engineering (SE) has witnessed an increase in interest in the industrial world due to systems becoming more and more complicated, particularly in the case of nuclear power plant design. Despite the fact that SE principles and procedures are now widely recognised, the challenge of moving from requirements to functional architectures has not yet been fully addressed. A three-pronged design perspective approach—a requirement view, a context view, and a behavioural view—is proposed in this work. Through the use of five connected Design Specific Modelling Languages (DSML), those views primarily assist the designers in expressing requirements, structuring their architecture designs, cooperating, verifying, and partially validating their designs.

[7] Carlos Hernández Corbato & Jose L Fernandez on Model-based systems engineering to design collaborative robotics applications, New robot technologies are becoming accessible to collaborate with people and automate jobs that are more sophisticated. To create and integrate this new generation of robotic systems, the automation and robotics industry requires methods and tools. In this study, the Model-Based Systems Engineering ISE&PPOOA technique is used to build robotic systems. The methodology is explained through the reengineering of a cutting-edge collaborative robot application. Model-based systems engineering's difficulties with robotic systems are explored, along with the advantages of MBSE techniques.

[8] Marvin Michael Manoury & Rainer Stark on Model-Based Systems Engineering (MBSE) as computer-supported approach for cooperative systems development, The need for cross-company and interdisciplinary collaboration is growing as a result of growing globalisation, the move towards Cyberphysical systems (CPSs), as well as smart products. To manage these systems' and goods' complexity A better version

of systems engineering, known as model-based systems engineering (MBSE), has arisen in engineering and is being embraced by many businesses. While attempting to keep up with current complexity trends, this technique only partially addresses the collaborative side of product creation. This paper will discuss how MBSE and engineering collaboration can be used to provide a computer-supported methodology for developing collaborative systems.

III. ROLE OF SYSML IN MBSE

Systems modelling language, or SysML, is a graphical modelling language used for model-based systems engineering (MBSE) applications. With the use of diagrams like requirements diagrams, use case diagrams, block definition diagrams, internal block diagram, activity diagrams, sequence diagrams, state machine diagrams, and parametric diagram among others, SysML diagrams, shown in Fig. 2, offers a standardised method for expressing complicated systems.

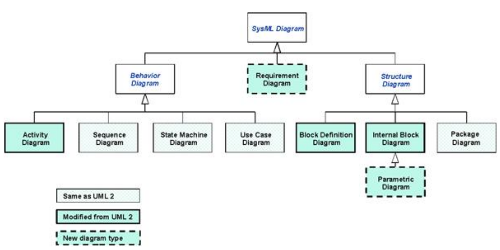

Fig. 2 SysML Diagrams

In MBSE, SysML is essential for creating a system model that encapsulates the needs, behaviour, structure, and interfaces of the system. Throughout the system's lifecycle, the SysML model serves as a blueprint for design, analysis, and verification.

Particularly, SysML enhances MBSE in the ways listed below:

1. Standardisation: SysML offers a standardised method of expressing systems, fostering uniformity, clarity, and stakeholder communication.
2. Modelling adaptability: SysML facilitates the modelling of a system's hardware and software components, allowing for a more thorough system model.
3. Management of needs: SysML provides modelling of system requirements, making it easier to connect requirements to design aspects.
4. Verification of the design: SysML models can be used to simulate and test the behaviour of the system, assisting in the early detection of design defects.
5. Communication and collaboration are made easier by the ability for team members and stakeholders to share SysML models.

In conclusion, SysML is a crucial tool in MBSE because it offers a standardised method for representing complex systems and makes it easier for stakeholders to collaborate and manage needs.

The four SysML pillars, which are shown in Fig. 3, are the fundamental ideas on which the language is built. They are as follows:

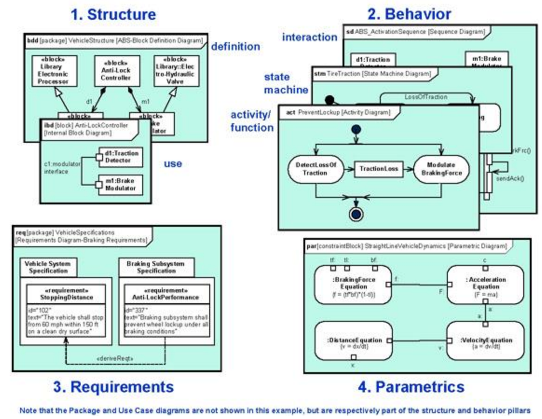

Fig. 3 Four Pillars of SysML

1. Structure: Blocks, pieces, ports, and connectors are just a few of the structural ideas offered by SysML to define a system's constituent parts and subsystems.
2. Behaviour: Using a range of diagrams, including activity diagrams, state machine diagrams, and sequence diagrams, SysML facilitates the modelling of system behaviour.
3. Requirements: SysML offers tools for managing and capturing system requirements, including attribution and tracability.
4. Parametrics: SysML provides the modelling of system constraints, equations, and other quantitative relationships through the use of parametric diagrams, allowing for the analysis and optimisation of system performance.

These four pillars work together to make it possible to build a complete system model that encapsulates a system's structure, behaviour, needs, and parametric relationships. This paradigm helps system stakeholders communicate and work together while acting as a potent tool for system design, analysis, and optimisation.

SysML Tool

Dassault Systèmes created the software suite CATIA MAGIC (Modelling and Generation of Integrated

Components), shown in Fig. 4, which includes the Systems of Systems Architect (SoS Architect) tool for model-based systems engineering (MBSE) using SysML. Using a modular and hierarchical approach, SoS Architect is a tool for creating and analysing complex systems of systems.

Fig. 4 CATIA Magic Systems of Systems Architect Icon

SoS Architect supports the four pillars of SysML, enabling the modeling of system structure, behavior, requirements, and parametric relationships. The tool provides a rich set of graphical modeling capabilities, including block diagrams, internal block diagrams, activity diagrams, sequence diagrams, state machine diagrams, and parametric diagrams.

One or more of SoS Architect's salient characteristics is:

1. Hierarchical modeling: Using SoS Architect, complex systems can be modelled hierarchically, including the division of system components into smaller parts and the assignment of requirements and restrictions to different system elements.
2. Management of system requirements: SoS Architect supports managing system requirements, including managing requirement modifications and tracing requirements to system components.
3. Analyses and simulation: SoS Architect enables trade studies and optimisation analyses as well as simulation and analysis of system behaviour and performance.
4. Collaboration: Collaboration is supported by SoS Architect, which enables system stakeholders to share system models and data as well as track changes and comments.

Overall, SoS Architect is an effective tool for MBSE utilising SysML that enables the development of detailed system models that accurately depict the structure, behaviour, specifications, and parametric linkages of complex systems of systems.

SysML Implementation

Every model in SysML starts with the package diagram. A SysML package diagram is a specific sort of diagram that demonstrates the hierarchy and connections between SysML packages. Blocks, interfaces, and diagrams are just a few examples of the SysML elements that can be contained in packages and used to organise related elements.

The organisation of packages in a SysML model is represented visually by the package diagram. Packages are shown as rectangles in a package diagram, and the relationships between them are shown as lines linking the rectangles. The diagram enables the modelling of complicated systems with several degrees of hierarchy by including packages at various levels of abstraction. The package diagram for the bricklaying robot is shown in the Fig. 5.

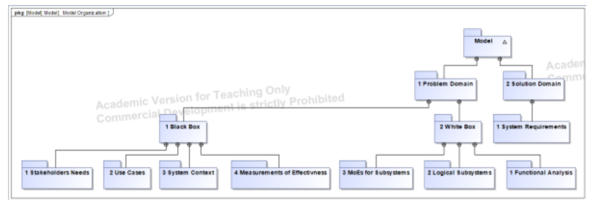

Fig. 5 Package diagram

The package diagram shows two main sections, which are Problem domain and Solution Domain. The whole systems model is contained inside the package diagram.

Problem Domain

The problem domain in SysML for the use of model-based systems engineering (MBSE) to support a bricklaying system in construction automation could be defined as follows:

1. System requirements: The problem domain begins with defining the system requirements, including performance, safety, and regulatory requirements.
2. Analysis of the construction process: The construction process must be analyzed to understand the requirements for the bricklaying system. This includes analyzing the bricklaying process, the materials used, and the environment in which the system will operate.
3. Identification of system components: The next step is to identify the components of the bricklaying system, including the bricklaying robot, the bricks, the mortar, and any sensors or controllers.
4. Definition of system interfaces: The interfaces between system components must be defined, including the inputs and outputs for each component.
5. Modeling of system behavior: SysML can be used to model the behavior of the system, including the sequence of actions taken by the robot to lay bricks, and the conditions under which the robot must operate.

6. Modeling of system structure: SysML can also be used to model the structure of the system, including the physical layout of the components and the connections between them.
7. Identification of system requirements and constraints: The requirements and constraints for the system components must be identified and allocated to the appropriate components.
8. Optimization and trade-off analysis: SysML can be used to perform trade-off analyses and optimization studies to identify the best design choices for the system.

By applying SysML to the problem domain of the bricklaying system in construction automation, the design team can create a comprehensive model of the system that captures all of the necessary requirements, constraints, and interactions between system components. This can help to identify and address design issues early in the development process, reducing costs and improving the quality of the final product.

a. Stakeholder Requirements Diagram

The relationships between stakeholders and their requirements are displayed in a stakeholder requirements diagram in SysML. It can aid in ensuring that the design satisfies the demands of all stakeholders and that their requirements are recorded in a stakeholder friendly language. Fig. 6 is an illustration of how a stakeholder needs diagram in SysML for a bricklaying robot may appear.

The design team can build a thorough model of the system that captures the relevant requirements, constraints, and interactions between system components by applying SysML to the problem domain of the bricklaying system in construction automation. This can assist in early design issue detection and resolution, lowering costs and raising the quality of the finished product.

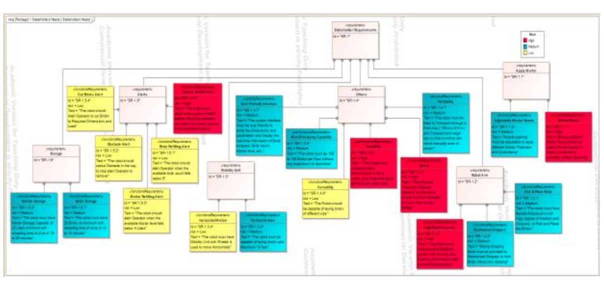

Fig. 6 Stakeholder Requirements Diagram

b. Use Case Diagram

A SysML use case diagram represents a system's functional needs from the viewpoint of the user. The actors (individuals or other systems that interact with the system) and use cases (tasks or actions that the system carries out) are often displayed. An illustration of a SysML use case diagram for bricklaying robot for construction automation with System Context of construction site is shown in Fig. 7.

You may visualise the system's anticipated behaviour and help to guarantee that all use cases are captured and taken into consideration in the system design by generating a use case diagram for a bricklaying system in construction automation. This can lower the chance of errors and guarantee that the system complies with all stakeholder criteria.

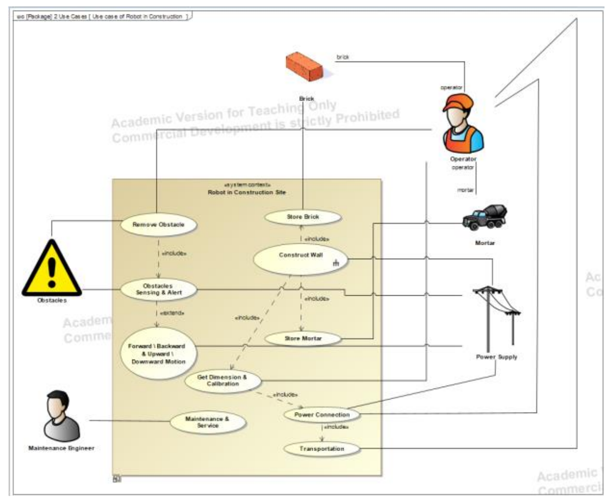

Fig. 7 Use Case Diagram

c. Activity Diagram

A SysML activity diagram is a diagram that shows how a system or process's activities or actions flow. It can be used to demonstrate the processes necessary to complete a certain activity or objective, and is often used to model the behaviour of a system. An illustration of a SysML activity diagram for bricklaying robot in construction automation is shown in Fig. 8.

For the bricklaying system in construction automation, you can model and visualise the behaviour of the system by developing an activity diagram in SysML. This can help you better understand how the system functions and pinpoint areas for improvement. By doing so, the possibility of errors may be decreased, and the system can be improved to meet the requirements and expectations of its users.

The activity diagram consists of two actors and the Bricklaying robot inside the horizontal swimlines. The swimlines constraints the activity carried by each actor with the system, thus helps the easy interpretation of the system and the related actor's behaviour. The activity diagram helps to develop the sequence diagram, which is derivation from activity diagram.

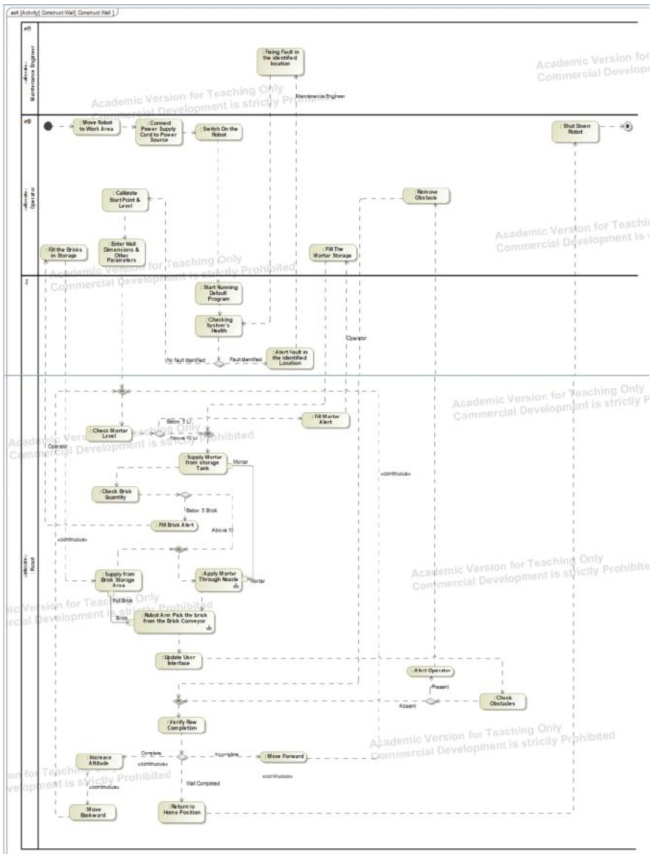

Fig. 8 Activity Diagram

d. Sequence Diagram

A SysML sequence diagram is a diagram that shows how various system elements interact with one another over time. It can be used to represent the flow of messages or actions between objects or components and is frequently used to simulate the dynamic behaviour of a system. An illustration of a SysML sequence diagram for bricklaying robot in construction automation is shown in Fig. 9.

You can model and visualise the interactions between items or components, which can help you better understand how the system functions and pinpoint areas for improvement, by generating a sequence diagram in SysML for the bricklaying system in construction automation. By ensuring that the system is tailored to the demands and expectations of its users, this can help to lower the likelihood of errors.

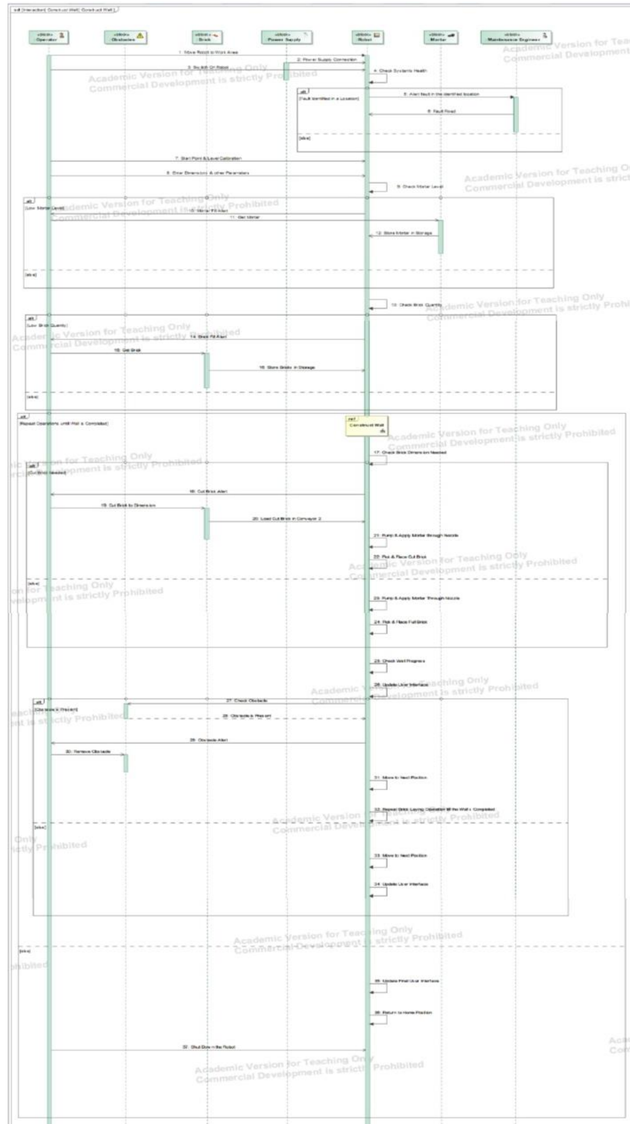

Fig. 9 Sequence Diagram

e. Block Definition Diagram

A graphical representation used to define the structure of a system or a component of a system is a SysML block definition diagram. It is used to depict a system's parts, as well as the characteristics of each part and the connections among them. The illustration of to a SysML block definition diagram of bricklaying robot in construction automation with robot structure and interfaces are shown in Fig. 10 & Fig. 11 respectively.

In order to better understand how the bricklaying system in construction automation functions and pinpoint areas for improvement, you can define the system's components and structure by creating a block definition diagram in SysML. By doing so, the possibility of errors may be decreased, and the system can be improved to meet the requirements and expectations of its users.

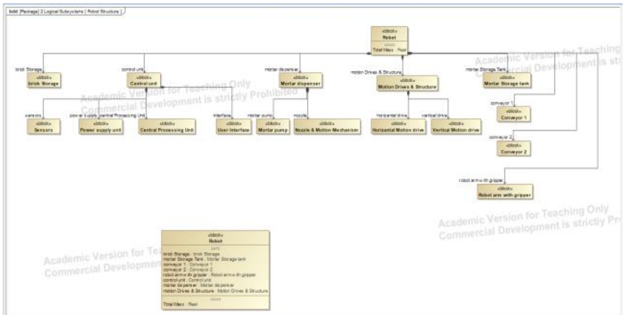

Fig. 10 Block Definition Diagram for Robot structure

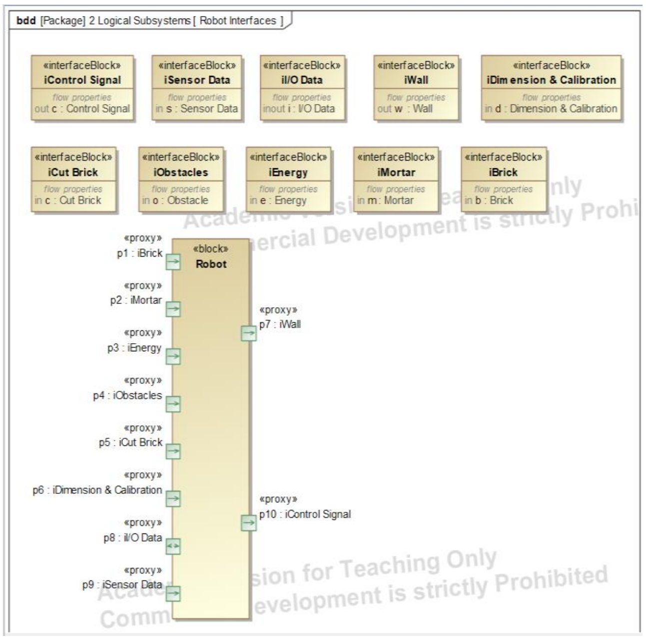

Fig. 11 Block Definition Diagram for Robot Interface

f. Internal Block Diagram

An internal block diagram in SysML is a visual representation of the internal organisation of a system or component of a system. It is utilised to represent a system's parts, their connections, and the transmission of information or control between them. The illustration of a SysML internal block diagram of bricklaying robot in construction automation is shown in Fig. 12.

By defining the internal structure of a system or a component of a system using an internal block diagram created in SysML, you can better understand how the system functions and pinpoint opportunities for improvement. By ensuring that the system is tailored to the demands and expectations of its users, this can help to lower the likelihood of errors.

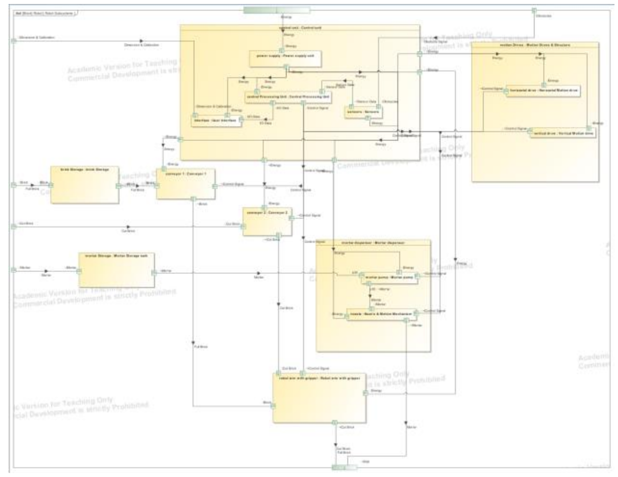

Fig. 12 Internal Block Diagram

Solution Domain

The solution domain in SysML is refinement of problem domain to support the design process with accurate required data. Generally, the solution domain consist of requirements, structure, behaviour and parametric of system, sub-system, and component level.

a. System Requirements Diagram

The system requirements for any system may consist of the followings,

functional requirements,
non-functional requirements,
physical requirements,
interface requirements,
performance requirements,
design constraints, and
verification requirements.

The above requirements must be satisfied with the design and development process of the system during the downstream engineering. The illustration of a SysML system requirements of bricklaying robot in construction automation is shown in Fig. 13.

The system requirements developed for bricklaying robot shows five design requirements to satisfy during development and testing phase. Designers can guarantee that the system satisfies the needs and expectations of its stakeholders and that it can be checked and validated to satisfy those criteria by expressing these requirements in SysML.

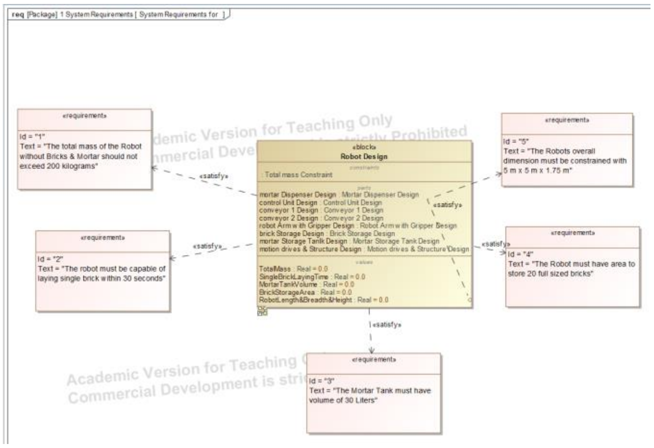

Fig. 13 System Requirements Diagram

b. State Machine Diagram

A graphical representation of the states, occurrences, and transitions of a system or subsystem is a SysML state machine diagram. It can be utilised to simulate a system's dynamic behaviour, demonstrating how it reacts to both internal and external events and how it evolves over time. The illustration of a SysML State Machine Diagram of bricklaying robot in construction automation is shown in Fig. 14.

The events "start", "bricks received", "bricks loaded", "bricks laid", and "job ended" cause the states to change. These occurrences may be brought about by internal sensors and control mechanisms inside the system itself, or external sources such as an operator or a construction management system.

This state machine diagram offers a high-level perspective of the behaviour of the system and can be used to make sure that the system operates as intended in a variety of scenarios and circumstances.

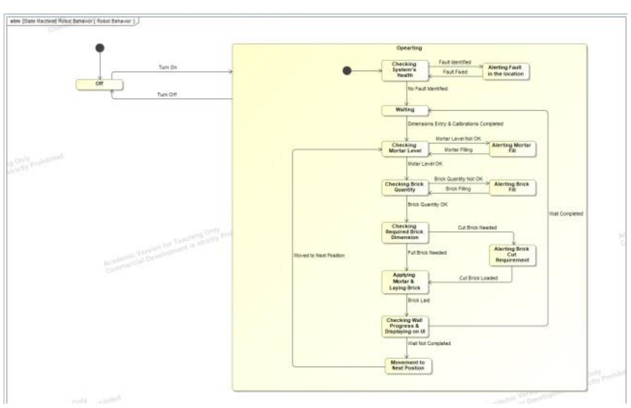
Fig. 14 State Machine Diagram

c. System Structure Diagram

A SysML system structure diagram is a graphical depiction of a system or subsystem's physical or logical components and the connections among them. It is used to represent a system's internal structure, including its parts, connections, and interfaces. The illustration of a SysML System Structure Diagram of Bricklaying Robot in construction automation with mass constraints for each sub-systems is shown in Fig. 15.

Connectors or lines are used to depict the connections between various elements. For instance, because the arm is a component of the actual robot, it is attached to the Robot Subsystem component. As they supply information to the robot's control system, the Sensors and Cameras components are similarly connected to the Control Subsystem component.

This system structure diagram offers a high-level perspective of the bricklaying system's elements and connections, and it may be used to check that the system is properly built and has all required elements and interfaces.

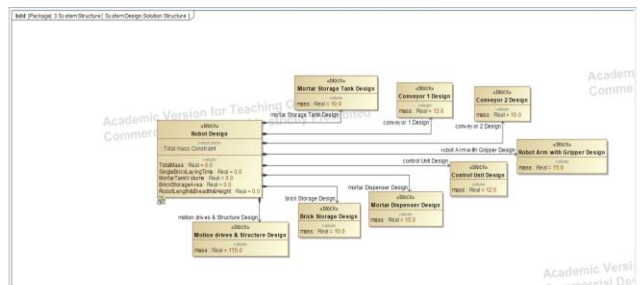
Fig. 15 System Structure Diagram

d. Parametric Diagram

A parametric diagram is used in SysML to model and examine the connections between a system's parameters, constraints, and numerical values. It is frequently employed to facilitate the understanding of a system's behaviour under various circumstances, such as modifications to inputs, outputs, or external factors.

A parametric diagram is made up of nodes that reflect a system's parameters, restrictions, values, and the connections between them. The relationships between the various nodes in the diagram are described by mathematical expressions and equations. The illustration of a SysML System Parameters Diagram and Parametric Diagram of Bricklaying Robot in construction automation with mass constraints for each sub-systems is shown in Fig. 16 and Fig. 17 respectively.

A parametric diagram could be used to model the relationship between the different system parameters in the case of the bricklaying system used in construction automation, including the speed and accuracy of the robot, the size of the bricks, and the possible effects of the environment (such as temperature and humidity) on the system's performance.

As well as equations and formulas that define the behaviour of the system under various settings, the parametric diagram may also incorporate limitations on the system, such as the maximum acceptable inaccuracy in brick placement and the maximum speed of the robot.

Overall, using a parametric model in SysML can help to make sure that the bricklaying system satisfies the desired performance standards and functions properly in various environmental settings.

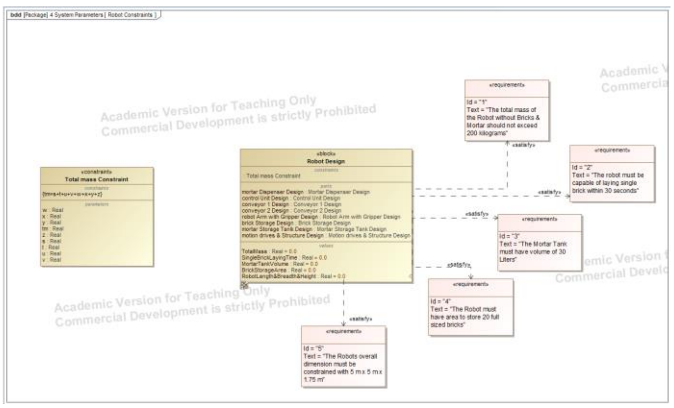

Fig. 16 System Parameters Diagram

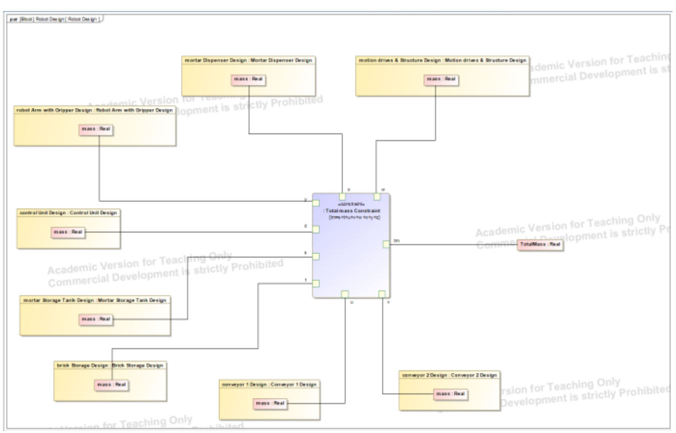

Fig. 17 Parametric Diagram

IV. RESULT & DISCUSSION

SysML is a useful tool for creating and analysing complicated systems, like the bricklaying system in construction automation, as it is a language for model-based systems engineering (MBSE). We can express the numerous facets of the system, such as its functional needs, physical components, and behaviour, in a straightforward and organised manner by utilising SysML to generate multiple models and diagrams.

The ability to model the many requirements of the system using various sorts of diagrams, such as use case diagrams, activity diagrams, and sequence diagrams, is one of the main benefits of utilising SysML for the design of the bricklaying system. These diagrams offer a clear overview of the system's operation, the various interactions between its parts, and the movement of information and control throughout the system.

Additionally, SysML offers the capability to model the physical structure of the system using block definition diagrams and internal block diagrams, allowing us to represent the various system components, their properties, and the relationships between them. This is particularly significant in the case of the bricklaying system, where the physical structure of the system, including the robot, the bricks, and the construction site, plays a critical role in the overall performance.

The ability to use state machine diagrams and parametric diagrams to model the behaviour of the system is another crucial component of the SysML implementation for the bricklaying system. While parametric diagrams offer a means to examine the interactions between the many parameters and constraints of the system, state machine diagrams assist in modelling the system's various states and transitions. These diagrams aid in ensuring that the bricklaying system satisfies the necessary performance standards and functions well in various environmental settings.

Overall, the SysML implementation for the building automation bricklaying system offers a potent tool for developing, deconstructing, and optimising the system. We can better comprehend the system's functionality, physical makeup, and behaviour by using SysML to model its various components. This will allow us to decide on the system's design and operation.

V. CONCLUSION

In conclusion, a solid and organised method for model-based systems engineering is provided by the use of SysML in the development of the bricklaying system in construction automation. SysML offers a precise representation of the system, including its needs, functionality, physical structure, and behaviour, through the numerous diagrams and models. This method enables system designers and engineers to carefully assess the system's performance and optimise it, guaranteeing that it satisfies the criteria and functions successfully under various environmental situations.

Additionally, the use of SysML in the development of the bricklaying system promotes improved stakeholder collaboration and communication, including that of designers, engineers, and end users. All stakeholders are on the same page and have a shared understanding of the system thanks to SysML's standardised language and notation, which improve understanding and communication of the system's requirements, design, and operation.

Because of this, the use of SysML in the design of the bricklaying system for construction automation can result in a more effective, efficient, and optimised system, which could ultimately result in more productivity, increased safety, and lower costs in the construction sector.

REFERENCES

[1] Zakaria Dakhli & Zoubeir Lafhaj | (2017) Robotic mechanical design for brick laying automation, Cogent Engineering, 4:1, 1361600.

[2] G. Pritschowa, M. Dalackera, J.Kurz a and M. Gaenssleb. Technological Aspects in the Development of a Mobile Bricklaying Robot, Automation in Construction, Volume 5, Issue 1, March 1996, Page 3-13.

[3] Hannu Lehtinen, Esko Salo, Heikki Aalto, Outlines of two Masonry Robot Systems, Technical Research Centre of Finland Vuorimiehentie 5, SF-02150 Espoo.

[4] Andrew J. Madsen, The SAM100: Analyzing Labor Productivity, Technical Research Centre of Finland Vuorimiehentie 5, SF-02150 Espoo.

[5] Aleksander Buczacki, An integrative Model Based Systems Engineering (MBSE) and lean-based approach for development of new complex products, MAPE 2019, volume 2, issue 1, pp. 425-434.

[6] L. Lemazurier, V. Chapurlat, A. Grossetête on An MBSE Approach to Pass from Requirements to Functional Architecture, IFAC Papers-Online, Volume 50, Issue 1, July 2017, Page 7260-7265.

[7] Carlos Hernández Corbato & Jose L Fernandez, Model-based systems engineering to design collaborative robotics applications, 2017 IEEE International Systems Engineering Symposium (ISSE).

[8] Marvin Michael Manoury & Rainer Stark, Model-Based Systems Engineering (MBSE) as computer-supported approach for cooperative systems development, 2020, European Society for Socially Embedded Technologies: vol. 4, no. 1

[9] Seung-Nam Yu, Byung-Gab Ryu, Sung-Jin Lim, Chang-Jun Kim, Maing-Kyu Kang, Chang-Soo Han, Feasibility verification of brick-laying robot using manipulation trajectory and the laying pattern optimization, Automation in Construction 18 (2009) 644–655.

[10] Juan Manuel Davila Delgado, Lukumon Oyedele* , Anuoluwapo Ajayi, Lukman Akanbi, Olugbenga Akinade, Muhammad Bilal, Hakeem Owolabi, Rotics and Automated systems in construction: Understanding industry specific challenges for adoption, Journal of Building Engineering 26 (2019), 100868.

[11] Aditya Akundi, Viviana Lopez, A Conceptual Model-based Systems Engineering (MBSE) approach to develop Digital Twins, 022 IEEE International Systems Conference (SysCon).

[12] Bjorn Colea , Vikram Mittalb,*, Stephen Gillespieb , Nguyen Lac , Richard Wisea , Alex MaxCalman, Model-based systems engineering: application and lessons from a technology maturation project, Procedia Computer Science 153 (2019) 202–209.