A Policy-Based Blackboard Control Mechanism for Dynamic Prioritization in Multilayered Problems

Hana Munira Muhd Mukhtar
Malaysian Institute of Information Technology, Universiti Kuala Lumpur, Kuala Lumpur, Malaysia
hanamunira@unikl.edu.my

Husna Sarirah Husin
School of Computer Science, Taylor's University, Subang Jaya, Malaysia
Husna.Husin@taylors.edu.my (corresponding author)

Azizah Rahmat
Malaysian Institute of Information Technology, Universiti Kuala Lumpur, Kuala Lumpur, Malaysia
azizah@unikl.edu.my

Received: 19 June 2025 | Revised: 25 July 2025 | Accepted: 11 August 2025

 DOI: https://doi.org/10.48084/etasr.12826

ABSTRACT

This research introduces a new blackboard control mechanism designed to address the weaknesses of conventional systems in managing complex, multilayered problem domains. Existing approaches are constrained by rigid, sequential control structures and limited scalability, making them unsuitable for environments that require adaptive and flexible decision-making. The proposed mechanism provides a standardized coordination framework capable of prioritizing several primary problems along with their associated subproblems. As a proof of concept, the control mechanism is applied to the domain of sustainable timber harvesting, a field characterized by interdependent tasks such as inventory assessment, growth prediction, ecological impact mitigation, and strategic tree selection. The mechanism supports dynamic problem prioritization according to a policy-based technique, allowing the system to adapt to progressing problem-solving needs. This study demonstrates the potential of an improved blackboard control mechanism to enhance decision support in complex, real-world domains.

Keywords-blackboard control mechanism; multilayered complex problem solving; dynamic prioritization; policy-based technique; sustainable timber harvesting

I. INTRODUCTION

Sustainable timber harvesting has become a crucial objective in modern forestry, driven by the need to preserve ecological balance while supporting continued economic viability. Innovations in sustainable forest management, particularly in harvesting techniques [1], aim to improve operational efficiency while adhering to sustainability principles. These approaches emphasize maintaining adequate timber yields, minimizing damage, supporting forest regeneration, and preserving the ecological structure of residual stands. These interrelated objectives create a complex, multilayered decision-making environment involving multiple distinct primary problems, each with associated subproblems that require independent, non-sequential decisions supported by diverse computational approaches. Furthermore, the processes, strategies, plans, and execution sequences for both primary problems and subproblems are unique, posing significant challenges for the control problem.

Therefore, sustainable timber harvesting is particularly well-suited as a proof-of-concept domain for a new mechanism capable of adaptive control. To operate effectively in such a dynamic context, expert systems must incorporate improved control mechanisms that support both standardized and adaptive inferencing. These systems must be able to compose complex, manageable problems and subproblems, prioritize them based on urgency and context, and integrate diverse knowledge. A systematic approach to knowledge design and control problem [2] enables such systems to reason across multiple decision layers, respond to changing priorities, and dynamic problem-solving conditions.

II. BACKGROUND STUDIES

An expert system simulates human expert decision-making using a knowledge base and inference engine. Multi-agent systems are commonly employed in expert systems [3]. The blackboard system, introduced in [4], is an extension of expert systems. The blackboard architecture, originally developed for the Hearsay-II project [5], has been effectively applied in diverse areas, including emergency response [6], robot task planning [7, 8], strategic automation [9-11], computational creativity in poetry [12], and cybersecurity operations [13-16].

The blackboard system consists of three main components [17]: independent knowledge sources, a shared workspace for centralized data sharing among KS, and a control mechanism that manages rule selection and KS execution. These components engage in an iterative cycle of proposing and refining partial solutions, allowing the system to construct outcomes in a flexible, step-by-step manner [18, 19]. The blackboard system features explicit knowledge source representation that differentiates between control and domain problems [6, 19, 20]. Control knowledge sources guide the problem-solving process, while domain knowledge sources supply the necessary expertise.

Benefiting from the explicit structure and separation, the blackboard system eases the composition of several primary problems along with their associated subproblems. In contrast, multi-agent systems may find it challenging to coordinate heterogeneous agents with distinct roles and knowledge, making integration increasingly difficult as system complexity grows [21-24]. This heterogeneity may complicate system design, particularly in managing interactions and communication [25, 26]. Conversely, blackboard systems offer centralized knowledge-sharing, enabling diverse knowledge sources to contribute effectively to a shared workspace [27]. Furthermore, the blackboard control mechanism offers a flexible, dynamic environment that supports various strategies like rule-based and heuristic methods [20]. It provides a solid framework that fosters innovation in managing complex, multilayered problems by enabling the comprehensive organization of knowledge sources and systematic control of problems.

III. ADDRESSING THE LIMITATIONS OF SUSTAINABLE TIMBER HARVESTING

This development responds directly to the limitations observed in current forest management practices, particularly Malaysia's long-established Selective Management System (SMS) [28-30]. SMS organizes operations into pre-harvesting, harvesting, and post-harvesting phases, but it exhibits several persistent shortcomings:

- Limited forest inventory (10% sampling).
- Generalized estimates of tree growth and felling damage.
- Manual directional felling based on logger expertise.
- Minimal post-harvest feedback integration.

Authors in [28, 30-35] have highlighted the consequences of these limitations, including skewed canopy gaps, slow forest regeneration, excessive volume loss, especially in non-dipterocarp species, and biodiversity degradation. A key underlying issue is the lack of precise, spatially referenced tree data, which impairs efforts to assess ecological impact, simulate growth trajectories, and optimize harvest strategies.

To overcome these challenges, the proposed technique incorporates:

- A detailed tree mapping database such as *x-y* coordinates, Diameter-Breast-Height (DBH), and species.
- Simulation-based growth forecasting tailored to forest type and geography.
- Directional felling analysis to minimize collateral damage and maximize yield.
- A reassessment mechanism to exclude trees where potential damage outweighs the production benefits.

In this study, a new blackboard control mechanism is introduced to address the limitations of sustainable timber harvesting. Sustainable timber harvesting is modeled as four primary, non-sequential problems: data preparation (P1), growth prediction (P2), minimum damage calculation (P3), and tree selection revision (P4). Each primary problem contains diverse, complex computational subproblems. The proposed solution represents these problems within well-defined domain knowledge sources to effectively manage the complexity of decision-making in a multilayered environment.

The control mechanism dynamically prioritizes both primary problems and subproblems based on factors such as complexity, execution time, and interdependencies. This adaptive approach enables flexible coordination of tasks in multilayered domain problems. The integration of structured knowledge sources and adaptive control contributes to more sustainable timber harvesting practices.

To realize the proposed solutions, a tree data collection study was conducted on January 14–15, 2020, in Compartment 17 of the Cherul Forest Concession, Dungun, Terengganu, in collaboration with Kumpulan Pengurusan Kayu Kayan Trengganu Sdn. Bhd. (KPKKT). A total of 48 trees from 21 species were sampled within a 300 m × 25 m plot, recording DBH, height, and *x-y* coordinates. The forest's degraded condition and small sample size made growth forecasting unfeasible. To address data limitations, a randomized technique for new tree recruitment [36] was adopted and translated into a computer simulation. Executed over repeated cycles, the simulation generated 22,934 trees across a 20-ha area (400 m × 500 m), with each tree defined by *x-y* coordinates, species name, species group, and DBH.

IV. CHALLENGES IN BLACKBOARD IMPLEMENTATIONS

Conventional blackboard systems, such as Hearsay-II [37], [38], OPM [4], and PREDMOLL [39], have provided a valuable framework for coordinating knowledge sources across modular tasks. However, their control structures typically fail to manage several primary problems with distinct subproblems, as required in sustainable timber harvesting. In particular, PREDMOLL was designed for protein structure prediction, exemplifying several limitations that highlight the need for a more adaptive control mechanism.

PREDMOLL is structured to solve a single main problem at a time, dividing it into two distinct phases: secondary structure prediction (e.g., α-helices, β-sheets) and tertiary structure prediction. Although these phases are logically interdependent, PREDMOLL enforces strict sequential execution. Each phase must be completed and the system reloaded before proceeding to the next, as there is no internal mechanism to transition or adapt between phases. This rigid workflow limits flexibility and adaptability in addressing real-world, multilayered problems.

PREDMOLL and similar systems do not support managing multiple primary problems with diverse subproblem selection processes or determining which primary problem should be addressed next. This limitation is especially problematic in the sustainable timber harvesting domain, where multiple primary problems may compete for attention, each requiring specific conditions for activation. The resolution process is inherently non-sequential; the selection and execution of subproblems vary across tasks and involve an additional layer of subproblems that further increase complexity.

V. METHODOLOGY

This methodology addresses the challenges associated with managing complex, multilayered problems composed of several independent primary problems and diverse subproblem structures [2]. The control mechanism integrates dynamic prioritization of primary problems with a policy-based technique for primary problem and subproblem selection and execution. Rather than following a fixed, sequential process, the system flexibly selects which primary problem to address based on the current state and urgency of the task. Once a primary problem is completed, the system reevaluates the problem space to determine the next most relevant primary problem, enabling an adaptive, non-linear progression aligned with current interest and state of problem-solving needs.

To guide this prioritization, a policy-based technique is embedded within the control mechanism. Each primary problem is evaluated according to predefined criteria such as complexity and execution time. This integrated approach provides a scalable control methodology, enabling expert systems to manage real-world scenarios where problems are interrelated, non-sequential, and layered with diverse subproblems.

VI. SYSTEM DESIGN

The proposed system is designed to manage complex, multilayered tasks by dynamically identifying and resolving problems in a flexible manner to meet problem-solving needs. Two key capabilities define its control mechanism:

- Dynamic prioritization of primary problems: This ensures that the system selects the next most relevant primary problem after completing the current one.
- Policy-based technique: the policy-based technique allows the system to handle the next primary problems and subproblems in a way that adapts to specific priorities such as processing time and computational complexity.

A. Dynamic Prioritization of Primary Problems

Each primary problem within the system contains its own domain-specific goal, such as predicting growth, calculating damage, or determining felling strategies in timber harvesting. Unlike rigid, sequential systems, this design updates its execution strategy dynamically based on the outcomes of previous cycles.

Building on earlier architectures, the standardized control process, as shown in Figure 1, follows a similar flow with key enhancements. It operates in three main steps involving four functions: trigger(), feasible(), scheduler(), and executeKS() to determine which knowledge source to execute within a standardized control loop.

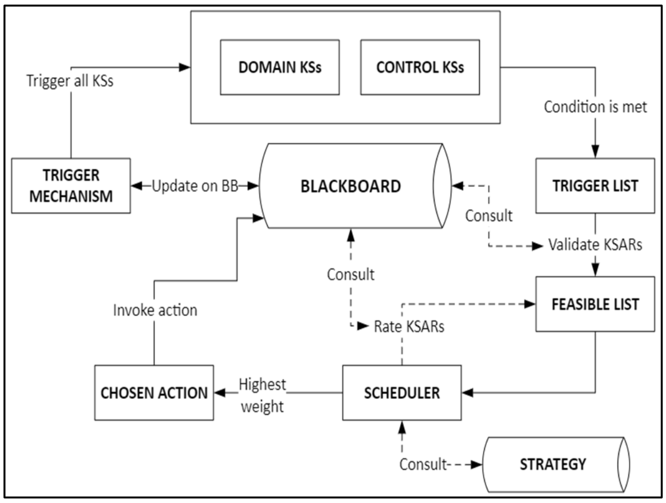

Fig. 1. Standardized control process.

Whenever the system finishes solving a primary problem, the blackboard updates the control event to 'Strategy End'. This signals that a new primary problem should be selected. At this point, the control knowledge source 'ksToSolveNewProblem' is activated. This follows a three-step control process to determine what should happen next. The following steps describe the activities performed, as presented in Table I:

B. Step 1: Update Event

1) Trigger()

- Identifies changes on the blackboard by initiating trigger conditions for all knowledge sources.
- If a condition is met, a Knowledge Source Activation Record (KSAR) is dynamically created and added to the trigger list.
- This step identifies potential knowledge sources for the problem-solving process.

2) Feasible()

- Corresponds to a specific event on the blackboard and subsequently validates all KSARs in the trigger list.
- The feasible condition is more specific than the trigger condition; once it is met, the KSAR is moved from the trigger list to the feasible list.

C. Step 2: Schedule KSARs

- KSARs in the feasible list compete for execution.
- The scheduler() function assigns weights to each KSAR based on current blackboard state, strategy status, domain level (w_{dl}), control level (w_{cl}), and policy criteria (w_p).
- The KSAR with the highest weight (w_{total}), indicating the best fit for the current context, is selected for execution.

D. Step 3: KSAR Execution

- The executeKS() function activates the selected KSAR, producing a new blackboard state that advances the problem-solving process.
- This action triggers the start of the next control cycle.

Table I shows how the weights (w_{dl}, w_{cl}, and w_p) are calculated for various KSARs, leading to the selection of 'ksToSolveNewProblem' due to its highest score. Upon selection, ksToSolveNewProblem determines that the current problem is PN, indicating the need to plan its strategy.

The planning process is tracked in the Blackboard (centralized data sharing among KS): Table II records PN as the new problem. Table III demonstrates how the ksPlanStrategy determined PN-related subproblems, and Table IV displays the transition to the Review Policy phase in Cycle 10.

As presented in Table I, the scheduler() rates all KSARs in the feasible list. For the current total weight of 720, columns w_{dl} and w_{cl} receive the current total weight of 720 accordingly. Given that the current blackboard next_control_level state is not Focus Domain, the control knowledge source ksToSolveNewProblem receives an additional weight of 100 for column w_p. The system chooses ksToSolveNewProblem as the selected action since it has the highest total weight value.

TABLE I. STANDARDIZED CONTROL PROCESS

Step 1: Update the event

	Condition	KSAR No.	KSAR
trigger( )	damage_tree.hit IS NULL	16	ksar('ksTreesInCircle', 'dip45,nondip50')
	blackboard.control_event =='Strategy End'	17	ksar('ksToSolveNewProblem', 'PN')
	prefelling.growth IS NULL	18	ksar('ksGrowthSimu', 22934)
	retain_tree.id IS NULL	19	ksar('ksEliminationECO', 30)

Step 2: Schedule the KSAR

	Trigger list	KSAR No.	Feasible list
feasible( )	ksar('ksTreesInCircle', 'dip45,nondip50')	16	ksar('ksTreesInCircle', 'dip45,nondip50')
	ksar('ksToSolveNewProblem', 'PN')	17	ksar('ksToSolveNewProblem', 'PN')
	ksar('ksGrowthSimu', 22934)	18	ksar('ksGrowthSimu', 22934)
	ksar('ksEliminationECO', 30)		

scheduler( )

KSAR No.	KS name	w_{dl}	w_{cl}	w_p	w_{total}
15	ksChangeFocus	0	620	100	720
16	ksTreesInCircle	720	0	0	720
17	ksToSolveNewProblem	0	720	100	820
18	ksGrowthSimu	720	0	0	720

Step 3: KSAR Execution

executeKS('ksToSolveNewProblem', 'PN')

TABLE II. DETERMINATION OF PROBLEM PN IN CYCLE 8 ksToSolveProblem

Cycle No.	KSAR No.	KS name	Problem	Domain level	Next control level	Control event
7	15	ksChangeFocus	P1	Preparation of data	Problem	Strategy End
8	17	ksToSolveNewProblem	PN	Problem	Plan Strategy	To Solve Problem

TABLE III. CREATION OF A NEW RECORD CYCLE 9 BY ksStartStrategyFocus

Cycle No.	KSAR No.	KS name	Problem	Domain level	Next control level	Control event
8	17	ksToSolveNewProblem	PN	Problem	Plan Strategy	To Solve Problem
9	20	ksPlanStrategy	PN	Problem	Focus	To Start Planned Strategy

TABLE IV. CREATION OF A NEW RECORD CYCLE 10 BY ksStartStrategyFocus

Cycle No.	KSAR No.	KS name	Problem	Domain level	Next control level	Control event
9	20	ksPlanStrategy	PN	Problem	Focus	To Start Planned Strategy
10	20	ksStartStrategyFocus	PN	Problem	Policy	Review Policy

E. Policy-Based Technique

Once a primary problem (e.g., PN) is defined, the system must then decide which primary problem to execute next. Instead of a static order, the system applies a policy-based technique that considers specific needs. The execution flow for PN identifies the primary problems P2, P3, and P4 as its subproblems by the ksPlanStrategy control knowledge source, as depicted in Table V. Initially, all subproblems are marked with a NULL status, indicating that their priority has not yet been assessed.

TABLE V. STORAGE OF PN SUBPROBLEMS BY f_strategy

No.	KS name	Status
1	P2	NULL
2	P3	NULL
3	P4	NULL

Next, ksStartStrategyFocus updates the status of each subproblem to To Review, as illustrated in Table VI, which triggers the ksProblemPolicy control knowledge source. This component examines each subproblem based on well-defined policy criteria, such as:

- Data Processing Time: How long it takes to compute a result.
- Calculation Complexity: How difficult the computational steps are.

The system applies specific policies to evaluate and prioritize subproblems using the control knowledge source ksProblemPolicy. As detailed in Table VII, these policies correspond to different types of problems and guide how subproblems should be selected for execution. The internal logic of this decision-making process is portayed in Figure 2, which outlines the process flow within ksProblemPolicy. According to the criteria defined in Table VIII, subproblem P2 is identified as faster and simpler to compute compared to P3, which involves a more elaborate three-step calculation, including damage assessment and felling direction analysis. Therefore, P2 is considered a high priority and is scheduled for execution.

TABLE VI. ksStartStrategyFocus UPDATES STATUS TO TO REVIEW

No.	KS name	Status
1	P2	To Review
2	P3	To Review
3	P4	To Review

TABLE VII. POLICY CONTROL FOR KNOWLEDGE SOURCES

KSAR No.	KS name	Problem	Domain level	Control level
2	ksProblemPolicy	PN	Problem	Policy
12	ksGrowthnYieldPolicy	P2S	Growth prediction	Policy
14	ksGrowthPolicy	P2	Growth prediction	Policy
19	ksEliminationPolicy	P4	Elimination P_LOG process	Policy

TABLE VIII. PROBLEM PRIORITIZATION CRITERIA

Criteria	Data processing time	Calculation complexity	Explanation
P2	Low	Low	P2 offers a growth and yield subproblem that iterates volume calculations based on tree diameter increments. P2's volume calculation is computationally simple.
P3	High	High	P3 requires a three-step complex calculation. The process involves three intricate steps: (1) determining the potential damage tree coordinates, (2) identifying the degree of potential damage, and (3) deciding the ideal felling direction for minimum damage volume and value. P3 involves multiple steps, requiring detailed computations for each harvest tree, significantly increasing processing time.

TABLE IX. ksProblemPolicy UPDATING P2 STATUS TO TO EXECUTE

No	Problem	Status
1	P2	To Execute
2	P3	Impractical
3	P4	Impractical

TABLE X. CREATION OF A NEW RECORD CYCLE 11 BY ksProblemPolicy

Cycle No.	KSAR No.	KS name	Problem	Domain level	Next control level	Control event
10	22	ksStartStrategyFocus	PN	Problem	Policy	Review Policy
11	23	ksProblemPolicy	PN	Problem	Focus	Change Focus

TABLE XI. CREATION OF A NEW RECORD CYCLE 12 BY ksChangeFocus

Cycle No.	KSAR No.	KS name	Problem	Domain level	Next control level	Control event
11	23	ksProblemPolicy	PN	Problem	Focus	Change Focus
12	27	ksChangeFocus	P2	Growth prediction	Plan	To Solve Problem

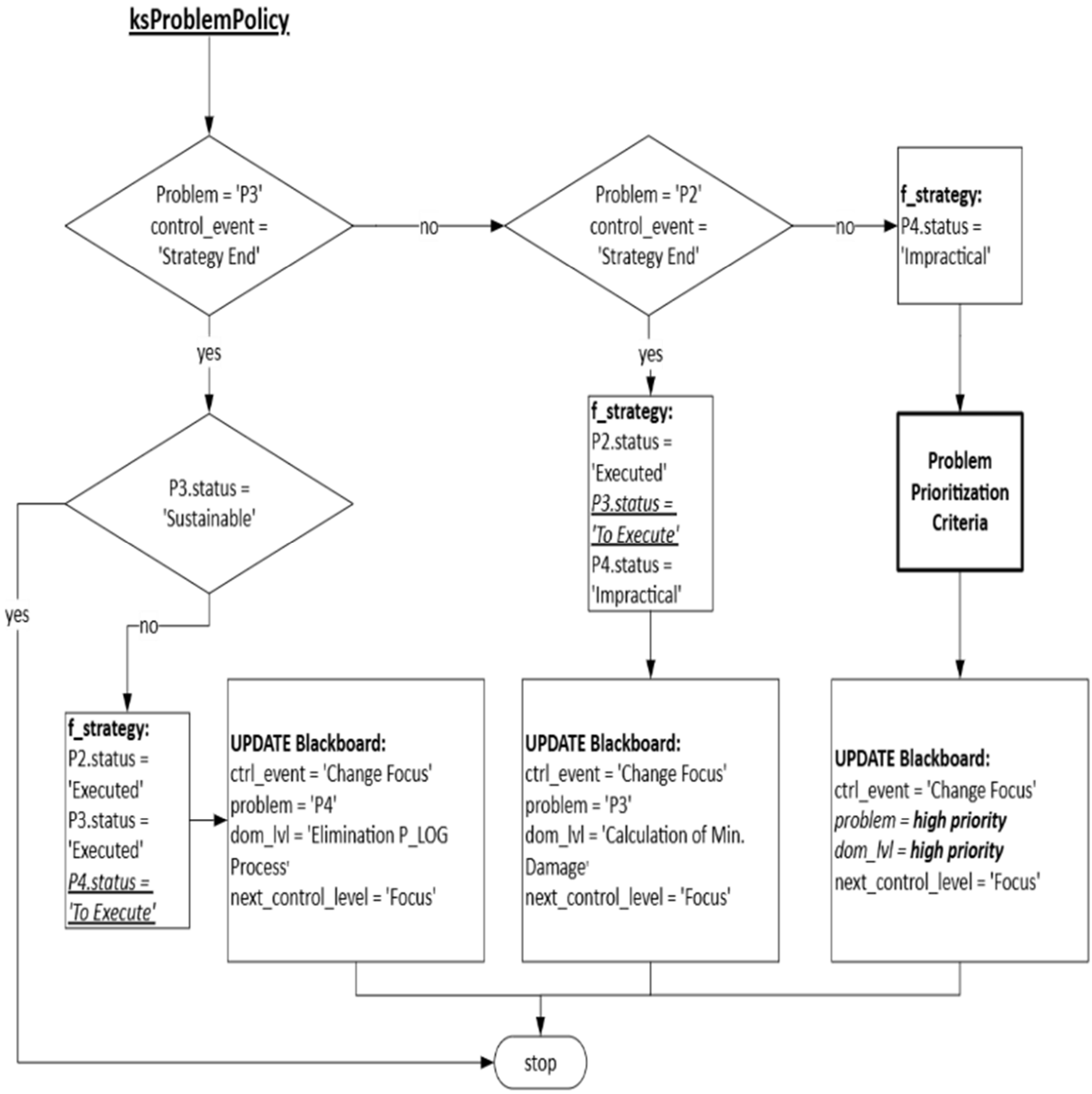

Fig. 2. ksProblemPolicy process flow.

Given this assessment, the system indicated that P2 is of higher priority and updated its status to To Execute while marking P3 and P4 as Impractical due to their higher computational demands and longer execution time under the current operational constraints. This prioritization decision is formally recorded in Table IX.

Following this, Table X presents the creation of a new cycle (Cycle 11), where the system's focus is shifted toward the execution of P2. To transition the system to P2, the control knowledge source ksChangeFocus is activated. This knowledge source finalizes the handover, and in the next cycle, Cycle 12, the system initiates the execution of P2, as documented in Table XI.

Through this policy-based technique, the system is able to make informed decisions that balance performance efficiency with problem complexity. This ensures that problems are not only solved in a timely manner but also in an order that maximizes overall system effectiveness.

VII. IMPLEMENTATION

The system is implemented as an improved blackboard control mechanism, where a central blackboard holds the current state, and knowledge sources are activated as conditions change. Each cycle on the blackboard captures an intermediate solution of decision-making.

A. *Dynamic Prioritization of Primary Problems*

The implementation begins with a control event, such as Strategy End, which triggers ksToSolveNewProblem. It is selected by evaluating which KSARs are most appropriate, based on total weighted scores.

- scheduler() rated KSARs (Table I), and the highest scoring one is selected.
- A new problem, PN, is identified (Table II).
- Planning phases are initialized across multiple cycles (Tables III and IV).

This ensures that the system adapts its priorities according to the current interest of problem needs, depending on which problems are already solved and what remains.

VIII. POLICY-BASED TECHNIQUE

During the policy review phase, the system evaluates subproblems using specific criteria. In the scenario shown:

- Subproblems P2–P4 are registered (Table V) and set to To Review (Table VI).
- ksProblemPolicy refers to the processing time and complexity to determine the high priority subproblem (Table VIII).
- P2 is chosen for execution, and ksChangeFocus updates the system state accordingly (Tables IX-XI).

Each decision updates the blackboard, triggering new control cycles and ensuring that the system remains responsive and policy-compliant throughout its execution.

IX. RESULT AND DISCUSSION

The proposed blackboard control mechanism effectively manages the complexity of sustainable timber harvesting by applying dynamic prioritization based on a policy-based technique. This allows the system to evaluate and schedule primary problems (P1–P4) and their subproblems according to computational complexity, dependencies, and execution time.

As illustrated in Figure 3, this flexibility contrasts sharply with conventional systems (Figure 4), which follow fixed, sequential processes and struggle with non-linear or iterative tasks such as regrowth prediction (P2) and tree selection revision (P4). The rigid nature of conventional blackboard systems limits adaptability and responsiveness in complex forestry scenarios.

By leveraging a policy-based technique, the proposed mechanism continuously reassesses task priorities, enabling more coherent decision-making and efficient coordination, essential for achieving sustainability goals in dynamic environments.

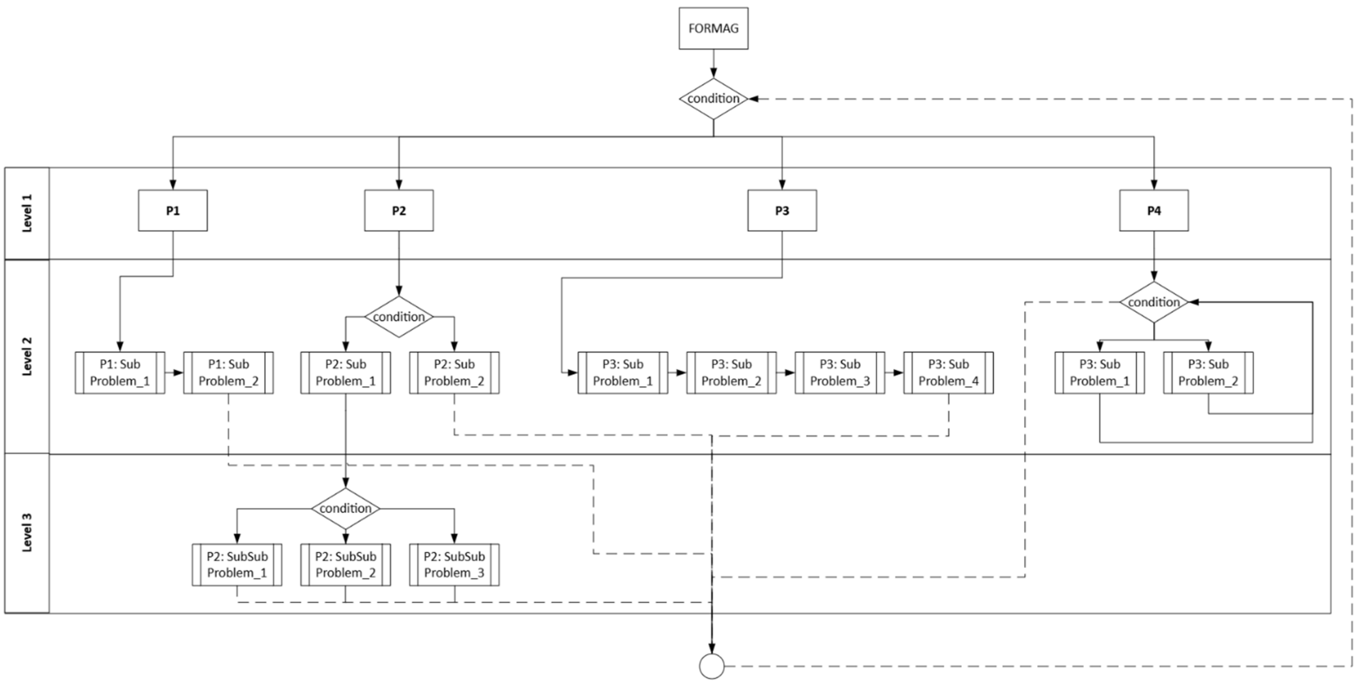

Fig. 3. Policy-based prioritization technique.

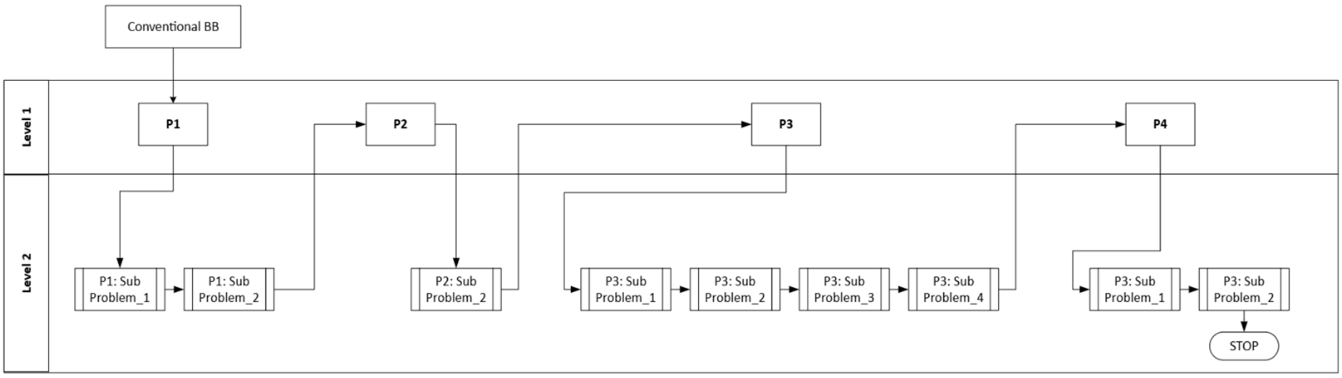

Fig. 4. Conventional blackboard in addressing sustainable timber harvesting problems.

X. CONCLUSION

This study presented a new blackboard control mechanism designed to overcome the structural and operational limitations inherent in conventional blackboard systems when applied to complex, multilayered problem domains. By incorporating problem prioritization according to a policy-based technique, the mechanism enables the adaptive selection and execution of both primary problems and their corresponding subproblems, taking into account current system states, computational complexity, and specific requirements.

Demonstrated within the context of sustainable timber harvesting, the proposed control framework successfully manages interdependent and non-sequential tasks such as growth prediction, damage assessment, and strategic tree selection. The mechanism's ability to re-evaluate priorities across problem-solving cycles ensures more coherent, efficient, and responsive decision-making.

The results affirm the mechanism's capacity to enhance system adaptability and scalability in environments where decision layers are intricately linked and continuously evolving. As such, this approach holds strong potential for broader application in other domains requiring intelligent coordination across dynamic, data-intensive, and policy-driven processes.

ACKNOWLEDGEMENT

The authors wish to express their sincere gratitude to Prof. Roslan Ismail, Dr. Yasmin Yahya, and Dr. Suriana Ismail for their invaluable supervision, guidance, and intellectual support throughout the development of this paper. The authors also acknowledge Taylor's University for providing funding for the publication of this paper, UniKL-MIIT for their continued support, and KPKKT for their technical expertise in conducting the tree data collection for this study.

REFERENCES

[1] H. M. Mukhtar, Y. Yahya, A. Rahmat, and R. Ismail, "Damage Cost/Value Clustering in Timber Harvesting Decision Making for Sustainable Forest Management," *International Journal of Electrical and Computer Engineering Systems*, vol. 12, pp. 31–39, Nov. 2021, https://doi.org/10.32985/ijeces.12.si.4.

[2] H. M. M. Mukhtar and R. Ismail, "The Design of Independent-Uniform Knowledge Sources of Blackboard Architecture in Timber Harvesting Decision-Making," in *2024 18th International Conference on Ubiquitous Information Management and Communication*, Kuala Lumpur, Malaysia, Jan. 2024, pp. 1–6, https://doi.org/10.1109/IMCOM60618.2024.10418421.

[3] S. T. Goonatilleke and B. Hettige, "Past, Present and Future Trends in Multi-Agent System Technology," *Journal Européen des Systèmes Automatisés*, vol. 55, no. 6, pp. 723–739, Dec. 2022, https://doi.org/10.18280/jesa.550604.

[4] B. Hayes-Roth, "A Blackboard Architecture for Control," *Artificial Intelligence*, vol. 26, no. 3, pp. 251–321, July 1985, https://doi.org/10.1016/0004-3702(85)90063-3.

[5] D. D. Corkill, "Blackboard Systems," *AI Expert*, vol. 6, no. 9, pp. 40–47, 1991.

[6] R. Hewett and S. Puangpontip, "On Controlling Drones for Disaster Relief," *Procedia Computer Science*, vol. 207, pp. 3703–3712, 2022, https://doi.org/10.1016/j.procs.2022.09.430.

[7] G. K. H. Pang, "Blackboard Architecture for Intelligent Control," *Control Systems, Robotics and Automation: Intelligent Control Systems*, vol. 17, pp. 303–316, 2009.

[8] R. Liscano, A. Manz, E. R. Stuck, R. E. Fayek, and J.-Y. Tigli, "Using a Blackboard to Integrate Multiple Activities and Achieve Strategic Reasoning for Mobile-robot Navigation," *IEEE Expert*, vol. 10, no. 2, pp. 24–36, Apr. 1995, https://doi.org/10.1109/64.395354.

[9] A. M. De Campos and M. J. Monteiro De Macedo, "A Blackboard Architecture for Perception Planning in Autonomous Vehicles," in *Proceedings of the 1992 International Conference on Industrial Electronics, Control, Instrumentation, and Automation*, San Diego, CA, USA, 1992, pp. 826–831, https://doi.org/10.1109/IECON.1992.254524.

[10] N. M. Sadeh, D. W. Hildum, T. J. Laliberty, J. McA'Nulty, D. Kjenstad, and A. Tseng, "A Blackboard Architecture for Integrating Process Planning and Production Scheduling," *Concurrent Engineering*, vol. 6, no. 2, pp. 88–100, June 1998, https://doi.org/10.1177/1063293X9800600201.

[11] S. B. Tor, G. A. Britton, and W. Y. Zhang, "A Knowledge-Based Blackboard Framework for Stamping Process Planning in Progressive Die Design," *The International Journal of Advanced Manufacturing Technology*, vol. 26, no. 7–8, pp. 774–783, Oct. 2005, https://doi.org/10.1007/s00170-003-2037-8.

[12] M.-R. Joanna and I. Bipin, "A Blackboard System for Generating Poetry," *Computer Science*, vol. 17, no. 2, 2016, Art. no. 265, https://doi.org/10.7494/csci.2016.17.2.265.

[13] J. Straub, "Modeling Attack, Defense and Threat Trees and the Cyber Kill Chain, ATT&CK and STRIDE Frameworks as Blackboard Architecture Networks," in *2020 IEEE International Conference on Smart Cloud*, Washington DC, WA, USA, Nov. 2020, pp. 148–153, https://doi.org/10.1109/SmartCloud49737.2020.00035.

[14] J. Rivard and J. Straub, "Extension of the Blackboard Architecture with Common Properties and Generic Rules." arXiv, 2023, https://doi.org/10.48550/ARXIV.2306.04287.

[15] J. Hance, J. Milbrath, N. Ross, and J. Straub, "Distributed Attack Deployment Capability for Modern Automated Penetration Testing,"

Computers, vol. 11, no. 3, Feb. 2022, Art. no. 33, https://doi.org/10.3390/computers11030033.

[16] J. Straub, "Application of Robotic Exploration Principles to the Challenge of Cybersecurity Penetration Testing," in *2024 IEEE International Conference on Cyber Security and Resilience*, London, United Kingdom, Sept. 2024, pp. 255–261, https://doi.org/10.1109/CSR61664.2024.10679404.

[17] P. Karl and B. Hayes-Roth, *An Introduction to Blackboard-Style Systems Organization*. Stanford, CA, USA: Knowledge Systems Laboratory, Stanford University, 1997.

[18] J. F. Fontanari, "Reputation Blackboard Systems," *Cognitive Systems Research*, vol. 50, pp. 29–35, Aug. 2018, https://doi.org/10.1016/j.cogsys.2018.03.008.

[19] A. Boaye Belle, T. C. Lethbridge, M. Garzón, and O. O. Adesina, "Design and Implementation of Distributed Expert Systems: On a Control Strategy to Manage the Execution Flow of Rule Activation," *Expert Systems with Applications*, vol. 96, pp. 129–148, Apr. 2018, https://doi.org/10.1016/j.eswa.2017.11.033.

[20] D. Janjanam, B. Ganesh, and L. Manjunatha, "Design of an Expert System Architecture: An Overview," *Journal of Physics: Conference Series*, vol. 1767, no. 1, Feb. 2021, Art. no. 012036, https://doi.org/10.1088/1742-6596/1767/1/012036.

[21] N. Krishnan, "Advancing Multi-Agent Systems Through Model Context Protocol: Architecture, Implementation, and Applications." arXiv, 2025, https://doi.org/10.48550/ARXIV.2504.21030.

[22] A. L. Laureano-Cruces, T. Ramírez-González, L. Sánchez-Guerrero, and J. Ramírez-Rodríguez, "Multi-Agent System for Real Time Planning Using Collaborative Agents," *International Journal of Intelligence Science*, vol. 04, no. 04, pp. 91–103, 2014, https://doi.org/10.4236/ijis.2014.44011.

[23] T. Pulikottil, L. A. Estrada-Jimenez, H. U. Rehman, J. Barata, S. Nikghadam-Hojjati, and L. Zarzycki, "Multi-Agent Based Manufacturing: Current Trends and Challenges," in *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation*, Vasteras, Sweden, Sept. 2021, pp. 1–7, https://doi.org/10.1109/ETFA45728.2021.9613555.

[24] N. Benmoussa, M. Fakhouri Amr, S. Ahriz, K. Mansouri, and E. Illoussamen, "Outlining a Model of an Intelligent Decision Support System Based on Multi Agents," *Engineering, Technology & Applied Science Research*, vol. 8, no. 3, pp. 2937–2942, June 2018, https://doi.org/10.48084/etasr.1936.

[25] F. Al-Shareefi, G. Chu, and A. Lisitsa, "A Multi-Agent Framework for Penetration Testing: Modelling and Analysing Using Abstract State Machines," in *2024 IEEE International Symposium on Parallel and Distributed Processing with Applications*, Kaifeng, China, Oct. 2024, pp. 1021–1028, https://doi.org/10.1109/ISPA63168.2024.00135.

[26] W. Jin, H. Du, B. Zhao, X. Tian, B. Shi, and G. Yang, "A Comprehensive Survey on Multi-Agent Cooperative Decision-Making: Scenarios, Approaches, Challenges and Perspectives." SSRN, 2025, https://doi.org/10.2139/ssrn.5106265.

[27] K. Chour *et al.*, "An Agent-based Modeling Framework for the Multi-UAV Rendezvous Recharging Problem," *Robotics and Autonomous Systems*, vol. 166, Aug. 2023, Art. no. 104442, https://doi.org/10.1016/j.robot.2023.104442.

[28] J. Jamhuri *et al.*, "Selective logging causes the decline of large-sized mammals including those in unlogged patches surrounded by logged and agricultural areas," *Biological Conservation*, vol. 227, pp. 40–47, Nov. 2018, https://doi.org/10.1016/j.biocon.2018.09.004.

[29] Y. M. Safiah Yusmah, M. J. Barnsley, and H. Rodziah, "Variation Analysis on Buffer Zone, Cutting-limit and Slope Angles in Timber Harvesting Planning," *Malaysian Forester*, vol. 73, no. 2, pp. 197–211, July 2010.

[30] I. Saiful and A. Latiff, "Canopy Gap Dynamics and Effects of Selective Logging: A Study in a Primary Hill Dipterocarp Forest in Malaysia," *Journal of Tropical Forest Science*, vol. 31, no. 2, pp. 175–188, May 2019, https://doi.org/10.26525/jtfs2019.31.2.175188.

[31] A. H. Atiqah, P. P. Rhyma, J. Jamhuri, A. W. Zulfa, M. S. Samsinar, and K. Norizah, "Using Google Earth Imagery to Detect Distribution of Forest Cover Change–is the Technique Practical for Malaysian Forests?," *Malaysian Forester*, vol. 83, no. 1, pp. 1–15, 2020.

[32] M. Azian *et al.*, "Carbon Emission Assessment from Different Logging Activities in Production Forest of Pahang, Malaysia," *Journal of Tropical Forest Science*, vol. 31, no. 3, pp. 304–311, Aug. 2019, https://doi.org/10.26525/jtfs2019.31.3.304.

[33] M. Demies, H. Samejima, A. K. Sayok, and G. T. Noweg, "Tree Diversity, Forest Structure and Species Composition in a Logged-over Mixed Dipterocarp Forest, Bintulu, Sarawak, Malaysia," *Transitions on Science and Technology*, vol. 6, pp. 102–110, Apr. 2019.

[34] N. J. Naim Jemali, M. F. Abd Rhani, M. Muhammad, and N. K. S. Abdul Majid, "Forest Growth Analysis of Ulu Sat Forest Reserve," *IOP Conference Series: Earth and Environmental Science*, vol. 549, no. 1, Aug. 2020, Art. no. 012039, https://doi.org/10.1088/1755-1315/549/1/012039.

[35] R. Pillay, F. Hua, B. A. Loiselle, H. Bernard, and R. J. Fletcher, "Multiple Stages of Tree Seedling Recruitment Are Altered in Tropical Forests Degraded by Selective Logging," *Ecology and Evolution*, vol. 8, no. 16, pp. 8231–8242, Aug. 2018, https://doi.org/10.1002/ece3.4352.

[36] Y. Yahya and R. Ismail, "Randomized Technique to Determine the New Seedlings for Simulation of Population Dynamic," in *Proceedings of the 13th International Conference on Ubiquitous Information Management and Communication 2019*, vol. 935, S. Lee, R. Ismail, and H. Choo, Eds. Cham, Switzerland: Springer International Publishing, 2019, pp. 711–722.

[37] V. Lesser, R. Fennell, L. Erman, and D. Reddy, "Organization of the Hearsay II Speech Understanding System," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, no. 1, pp. 11–24, Feb. 1975, https://doi.org/10.1109/TASSP.1975.1162648.

[38] V. R. Lesser and L. D. Erman, "A Retrospective View of the Hearsay-II Architecture," in *Proceedings of the 5th international joint conference on Artificial intelligence*, 1977, pp. 790–800.

[39] R. B. Ismail, "The Use of Artificial Intelligence Techniques for Protein Structure Prediction," Ph.D. dissertation, University of Glasgow, Glasgow, United Kingdom, 1989.