PRIORITIZING PERFORMANCE EVALUATION FACTORS OF EVENT-BASED INFORMATION SYSTEMS USING INTERPRETIVE STRUCTURAL MODELING

Mangesh Joshi

Received: April 15, 2023; Revised: June 04, 2023; Accepted: June 08, 2023

Abstract

This paper is aimed at the application of Interpretive Structural Modeling (ISM) for prioritizing the factors associated with the performance evaluation of event-based information management system (EBIMS). The study identified thirteen such critical factors deciding the performance of Event-Based Information Systems. Literature review along with experts' opinions were collected to arrive at the final thirteen factors. In this paper, the authors have used Interpretive Structural Modeling (ISM) approach to interpret the interdependency among the selected factors. In addition, MICMAC (cross-impact matrix multiplication applied to classification) analysis is also performed to illustrate the relative driving and dependence power among the selected factors. This paper infers that event processing algorithm, data volume and quality, and hardware and software along with query complexity are the most dominating factors which have the highest driving power and the minimum dependence power as they drive other factors and sit at the top of the interpretive structure model.

Keywords: **Interpretive Structural Modeling; MICMAC analysis, Performance evaluation, event based information management**

Introduction

An event-based management system (EBMS) is a software system that manages events, which are typically defined as a change of state or the occurrence of a significant activity in a business process (Cugola *et al*., 2001). This type of system has become increasingly popular in recent years due to the growing demand for real-time data processing and analysis. EBMS operates by capturing events from various sources, such as sensors, applications, and databases, and then processing them in real-time to identify patterns, correlations, and trends. The system can then trigger appropriate actions, such as sending alerts, generating reports, or initiating workflows, based on the analysis of the events. One of the key advantages of EBMS is that it enables businesses to react quickly to changes in their environment, such as market trends, customer behavior, or production processes. By providing real-time insights into events, businesses can make more informed decisions and take proactive

Shri Ramdeobaba College of Engineering and Management, Katol Road, Nagpur, MH, India. E-mail:joshimp@rknec.edu
* *Corresponding author*

measures to address issues before they escalate. Another advantage of EBMS is that it can be used to improve operational efficiency and reduce costs. By automating the management of events, businesses can reduce manual intervention and streamline processes, resulting in faster response times and reduced errors.

However, designing and implementing an EBMS can be complex, and requires careful consideration of a range of factors, including the types of events to be managed, the volume and frequency of events, the latency requirements, and the scalability of the system. An event-based management system is a powerful tool for businesses seeking to gain real-time insights into their operations and take proactive measures to address issues. By leveraging the latest advances in event processing and analysis, businesses can improve their operational efficiency, reduce costs, and stay ahead of the competition.

Various event-based applications can be found in the literature. A system was developed for processing data based on events that utilizes a Kafka container cluster within a Kubernetes environment (Liu *et al*., 2023). Authors (Hinze *et al*., 2009) highlighted seventeen applications of event-based systems in their research. It includes Content awareness: Mobile Tourist Information System, Baggage Management, Traffic Monitoring, Environmental Monitoring: Avalanche Warning System, Information Dissemination, Blog Information Mining & Dissemination, Fraud Detection, Financial Applications, Supply Chain Management, Piping Information to the Cockpit, Railway Scenario – Cargo, Health Care: Health Monitor, Smart Cities and Infrastructure for Ambient Intelligence, Ambient Assisted Living, Smart Homes, Autonomic Computing, Self-X Systems, Threat Detection (Dirty Bomb, Spills), and Gaming.

Why Interpretive Structural Modeling ?

Interpretive Structural Modeling (ISM) is a technique that enables the development of a hierarchical model representing the interrelationships among various factors in a complex system. The technique was first proposed by Warfield (1974) and has since been used in numerous applications. The process involves a series of steps that help to identify the relationships between different elements of the system, and to identify the most important elements that affect the system's overall performance. ISM has been used in various domains to analyze complex systems. One of the key benefits of using ISM is that it can help researchers identify the key drivers of a system's performance. By identifying these drivers, researchers can better understand the factors that are most important in determining the overall performance of a system. Recent studies have demonstrated the utility of ISM in a range of domains (Joshi and Deshpande, 2022).

As each study identifies different significant factors, it is important to establish the relationship between these factors. Therefore, this paper will apply ISM in prioritizing the performance factors in the further part of this paper.

Literature Review

To determine the factors that impact the performance of event-based systems, a literature review was conducted using Scopus, Web of Science, and Google Scholar indexed journals. The search utilized keywords such as ‘performance’, ‘event-based system’, ‘continuous event processing’, ‘event-based architecture performance’, ‘system design factors’, and ‘system design considerations’. Through this review, a list of contributing factors (CFs) was identified.

Figure 1(a) shows the systematic methodology for selection of factors which was followed in this study. It begins with selection of experts. A total of eight experts consisting of three from industry and five from academia was chosen. A discussion was held with a group and all previously identified factors were presented. They were asked to add missing factors if any. A questionnaire was prepared and given to individual expert for pairwise comparison of all factors. The responses were received and analyzed. The ranks obtained for the factors were shared among the experts. The final ranking with weights were obtained after discussion between the experts. This discussion led to the identification of thirteen relevant and important CFs for this research. The detailed step wise result of factor selection methodology is given in Figure 1(b).

Performance evaluation of event-based management systems (EBMS) is an essential aspect of ensuring that the system can handle the demands of real-time data processing and analysis. This literature review explores the key factors that affect the performance of EBMS and the methods used to evaluate their performance.

- Performance requirements like high throughput and/or low latency

- Operational requirements throughput, response time, type of events, number of sources, number of sinks, scalability, and many more.

The factors which were finalized are describred as following:

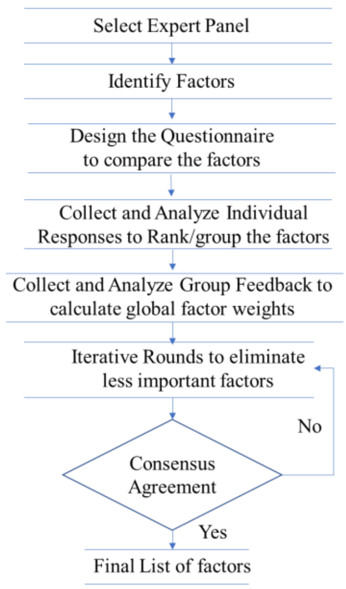

Figure 1(a). Factor selection methodology

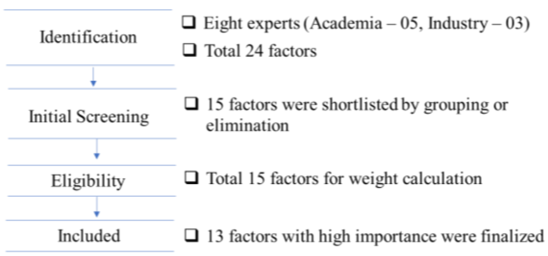

Figure 1(b). Result of Factor selection

1.1 Event processing algorithms

The algorithms used to process events are a critical factor that can impact the performance of an EBIM system. Complex algorithms may require more processing power, while simpler algorithms may not provide enough accuracy or detail. Additionally, the efficiency of the algorithms can impact the speed of processing events, leading to delays in real-time data processing. Choosing the right algorithms and optimizing them can be critical to ensuring the performance of the EBIM system (Liu *et al*., 2023; Long and Jia, 2023).

1.2 Data volume and Quality

The volume of data generated by an EBIM system can also impact its performance (Liu *et al.*, 2023). As the volume of data increases, the system may become slower and less responsive, and may require more resources to process the data. According to Fisher *et al.* (2022), the increasing volume and variability of data generated from industrial bioprocesses make it challenging to develop data-driven models that accurately capture the underlying biological nature of the bioprocesses. The quality of the data generated by the system is critical to the accuracy and usefulness of the EBIM system.

1.3 Hardware and software

The hardware and software used to run the system can impact performance (Prasadana, 2022). Outdated hardware or software may not provide enough processing power or memory, while newer hardware or software may require more resources than the system can provide. Ensuring that the hardware and software are compatible and optimized for the system's requirements is critical for optimal performance.

1.4 Network latency

The latency of the network used to transmit events can impact performance. High latency can result in delays or dropped events, while low latency can provide faster and more accurate event processing. Ensuring that the network infrastructure is optimized to reduce latency can help improve system performance (Mendes *et al*., 2009; Ren, *et al*., 2023).

1.5 Memory management

Monitoring and managing the system can help identify and resolve performance issues. Real-time monitoring can help identify bottlenecks or errors, while proactive management can help prevent issues from occurring in the first place. Ensuring that the system is properly monitored and managed can help improve system performance (Flouris *et al*., 2017).

1.6 System scalability

Scalability of the system is also an important factor that can impact performance (Ciociola *et al.*, 2023). A system that is not designed to scale may become overwhelmed with increased load, resulting in degraded performance. Designing the system with scalability in mind, such as using distributed computing or containerization, can help ensure that the system can handle increasing loads without impacting performance.

1.7 Fault Tolerance

An EBMS should be able to handle failures gracefully and continue to operate without disruption. The system should be designed with fault tolerance in mind and should be able to recover from failures quickly and without data loss (Li *et al*., 2023; Yang *et al*., 2023).

1.8 Throughput / Event Rate

Throughput/Event Rate is a measure of the number of events that the system can process per unit time. A high throughput is essential in ensuring that the system can handle large volumes of events and respond quickly to changing business requirements (Mendes *et al*., 2009).

1.9 Traceability

Traceability is also an important aspect in event-based systems (Cleland-Huang *et al*., 2003). Since event-based systems rely on the exchange of events between different components or services, it is important to be able to trace the events and their flow throughout the system. It is particularly important in scenarios where the system needs to meet certain regulatory or compliance requirements. For example, in financial systems, traceability is required to ensure that all transactions are properly recorded and audited for compliance with regulations.

1.10 Window Policies

A window policy refers to a set of rules that determines how events are grouped and processed over a specific time interval or event count. Window policies are used to manage the flow and processing of real-time data streams, and to identify and analyse patterns and trends within the data (Mendes *et al*., 2009).

1.11 Number of sources and sinks

Sources and sinks refer to the inputs and outputs of the CEP system. Sources are the streams of data that are input into the CEP system for analysis, while sinks are the destinations for the results of the analysis. The number of sources and sinks in a CEP system can vary depending on the specific use case and application requirements (Cugola *et al*., 1998).

1.12 Queries/patterns complexity

It refers to the rules or conditions used to detect specific events or event sequences in real-time data streams. The complexity of queries and patterns in CEP can vary depending on the number of input streams, the number of conditions or rules, and the types of operations involved (Mendes *et al*., 2009).

1.13 Event type

In CEP, events are typically categorized into different types based on their properties and characteristics (Boer *et al*., 2004). The type of event is determined by the nature of the data it contains, and can be used to filter, group, and analyse events in different ways. Some includes simple, composite and complex events.

The performance of an event-based information management system is impacted by a variety of factors. From the algorithms used to process events to the behaviour of users interacting with the system, each factor plays a role in determining the system's responsiveness and accuracy. By optimizing these factors and balancing performance with data quality and system requirements, businesses can ensure that their EBIM system is performing at its best.

Materials and Methods

The ISM approach is based on the assumption that the components of a system are interdependent and that their relationships can be modelled as a hierarchy of levels. Each level represents a set of components that are related to each other and have a similar level of importance within the system. The components at the top of the hierarchy are the most important, while those at the bottom are the least important.

The construction of a logical mental diagram using Interpretive Structural Modeling (ISM) involves a systematic methodology. Figure 2 illustrates the following steps:

1. Identify factors that are relevant to the specific problem through an extensive literature survey, expert opinion, or group problem-solving technique. The list can then be finalized through group decision-making.
2. Establish contextual relationships among the finalized contributing factors.
3. Develop a Structural Self-Interaction Matrix (SSIM) that shows the pairwise relationship between the problem factors.
4. Create a reachability matrix from the SSIM and check for transitivity. The assumption of transitivity of contextual relations is fundamental in ISM. It implies that if factor X is related to Y and Y is related to Z, then X is related to Z.
5. Partition the reachability matrix into different levels.
6. Draw a directed graph (DIAGRAPH) based on the relationships in the reachability matrix and remove transitive links.
7. Replace factor nodes with statements to convert the resulting diagraph into an ISM.

8. Due to the complex structure of the problem, conceptual inconsistencies may arise. Therefore, reviewing is necessary to identify and make any necessary modifications to the ISM.

Structural self-interaction matrix (SSIM)

According to the ISM methodology, the next step involves utilizing expert opinions to establish contextual relationships between factors. To gather these opinions, an Excel-VBA macro based front end (as depicted in Figure 3) was created and distributed among the experts. They were asked to compare a single factor with another factor and indicate the relationship between them by pressing one of four buttons (V, A, X, or O). These relationships were then analyzed using four variables to determine the direction of the relationship between the two factors (i and j) in the development of SSIM. The direction of the relationship between two factors (i and j) in the development of SSIM is determined using four variables: V, which indicates that Factor i influences Factor j; A, which indicates that Factor i is influenced by Factor j; X, which indicates that Factors i and j will help achieve each other; and O, which indicates that Factors i and j are unrelated.

The experts were instructed to provide remarks or comments in the remark box explaining the reasoning behind the established relationship. Once the filled Excel sheets were collected, any discrepancies were discussed and resolved. Based on the final assessment of the experts, a contextual relationship for SSIM was created and is presented in Table 1.

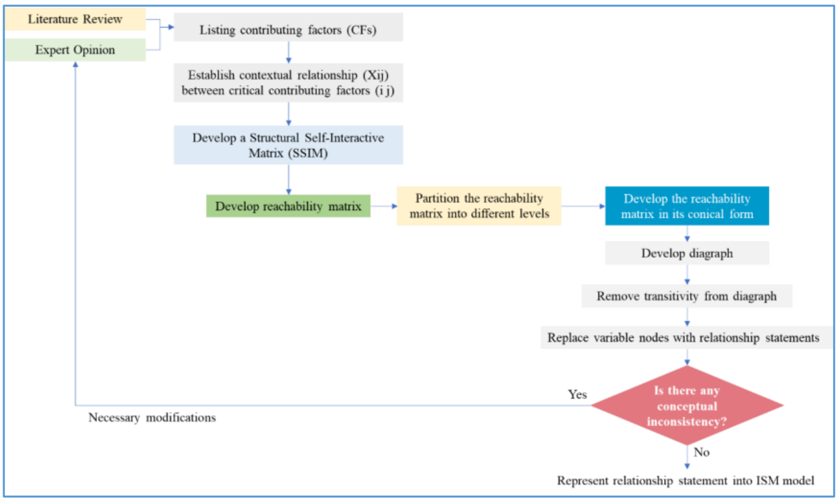

Figure 2 **Research Methodology (Source: Joshi and Deshpande, 2022)**

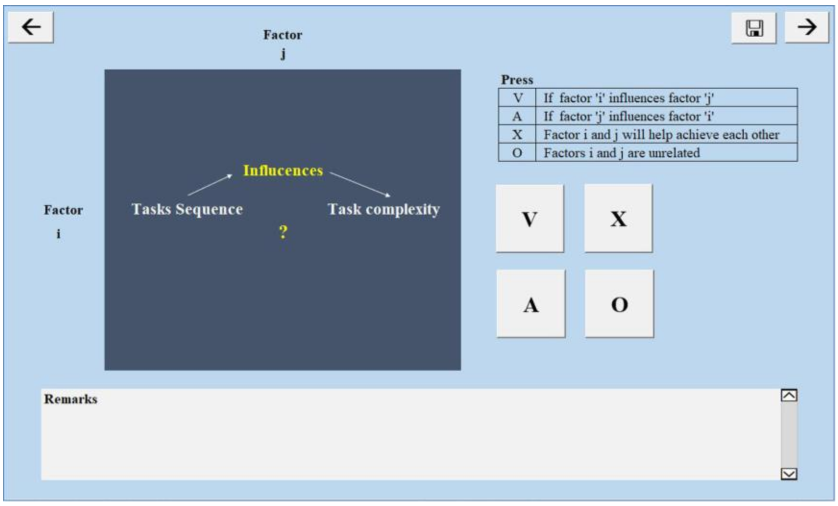

Figure 3 **Screenshot of front end**

Table 1. Structural self-interaction matrix (SSIM)

FN	Factors/problems	1	2	3	4	5	6	7	8	9	10	11	12	13
F1	Event processing algorithms	-	A	A	V	V	V	V	V	V	V	O	V	A
F2	Data volume and quality		-	V	V	V	A	V	V	O	V	V	V	O
F3	Hardware and software			-	V	V	V	O	V	V	A	O	A	A
F4	Network latency				-	A	O	V	V	V	V	A	A	A
F5	Memory management					-	V	O	V	V	A	A	A	A
F6	System scalability						-	O	O	O	A	V	A	A
F7	Fault Tolerance							-	V	A	O	O	A	A
F8	Throughput / Event Rate								-	A	A	A	A	V
F9	Traceability									-	O	O	A	A
F10	Window Policies										-	V	O	O
F11	Number of sources and sinks											-	V	V
F12	Queries/patterns complexity												-	O
F13	Event Type													-

Table 2. Initial Reachability Matrix (IRM)

FN	Factors/problems	1	2	3	4	5	6	7	8	9	10	11	12	13	DR.P.
F1	Event processing algorithms	1	0	0	1	1	1	1	1	1	1	0	1	0	**9**
F2	Data volume and quality	1	1	1	1	1	0	1	1	0	1	1	1	0	**10**
F3	Hardware and software	1	0	1	1	1	1	0	1	1	0	0	0	0	**7**
F4	Network latency	0	0	0	1	0	0	1	1	1	1	0	0	0	**5**
F5	Memory management	0	0	0	1	1	1	0	1	1	0	0	0	0	**5**
F6	System scalability	0	1	0	0	0	1	0	0	0	0	1	0	0	**3**
F7	Fault Tolerance	0	0	0	0	0	0	1	1	0	0	0	0	0	**2**
F8	Throughput / Event Rate	0	0	0	0	0	0	0	1	0	0	0	0	1	**2**
F9	Traceability	0	0	0	0	0	0	1	1	1	0	0	0	0	**3**
F10	Window Policies	0	0	1	0	1	1	0	1	0	1	1	0	0	**6**
F11	Number of sources and sinks	0	0	0	1	1	0	0	1	0	0	1	1	1	**6**
F12	Queries/patterns complexity	0	0	1	1	1	1	1	1	1	0	0	1	0	**8**
F13	Event Type	1	0	1	1	1	1	1	0	1	0	0	0	1	**8**
	DEPENDENCE POWER	**4**	**2**	**5**	**8**	**8**	**7**	**7**	**11**	**7**	**4**	**4**	**4**	**3**	**74**

Table 3. Final Reachability matrix

FN	Factors/problems	1	2	3	4	5	6	7	8	9	10	11	12	13	DR.P.
F1	Event processing algorithms	1	1	1	1	1	1	1	1	1	1	1	1	1	13
F2	Data volume and quality	1	1	1	1	1	1	1	1	1	1	1	1	1	13
F3	Hardware and software	1	1	1	1	1	1	1	1	1	1	1	1	1	13
F4	Network latency	0	0	1	1	1	1	1	1	1	1	1	0	1	10
F5	Memory management	0	1	0	1	1	1	1	1	1	1	1	0	1	10
F6	System scalability	1	1	1	1	1	1	1	1	0	1	1	1	1	12
F7	Fault Tolerance	0	0	0	0	0	0	1	1	0	0	0	0	1	3
F8	Throughput / Event Rate	1	0	1	1	1	1	1	1	1	0	0	0	1	9
F9	Traceability	0	0	0	0	0	0	1	1	1	0	0	0	1	4
F10	Window Policies	1	1	1	1	1	1	0	1	1	1	1	1	1	12
F11	Number of sources and sinks	1	0	1	1	1	1	1	1	1	1	1	1	1	12
F12	Queries/patterns complexity	1	1	1	1	1	1	1	1	1	1	1	1	1	13
F13	Event Type	1	1	1	1	1	1	1	1	1	1	1	1	1	13
	DEPENDENCE POWER	9	8	10	11	11	11	12	13	11	10	10	8	13	**137**

Reachability matrix

To transform SSIM into a two-dimensional matrix known as the reachability matrix, V, A, X, and O were replaced with either 1 or 0 based on specific conditions. The substitution of 1 and 0 followed the following rules:

(i) If CellSSIM(i, j) == ‘V’, then CellRM (i, j) is set to 1 and CellRM (j, i) is set to 0.

(ii) If CellSSIM(i, j) == ‘A’, then CellRM (i, j) is set to 0 and CellRM (j, i) is set to 1.

(iii) If CellSSIM(i, j) == ‘X’, then CellRM (i, j) is set to 1 and CellRM (j, i) is set to 1.

(iv) If CellSSIM(i, j) = ‘O’, then CellRM (i, j) is set to 0 and CellRM (j, i) is set to 0.

According to the aforementioned guidelines, the initial reachability matrix (IRM) was constructed as shown in Table 2. After implementing the transitivity rule, the final reachability matrix was generated and can be seen in Table 3. Furthermore, Table 3 exhibits the driving and dependence power for each factor. The driving power refers to the number of factors influenced by a particular factor, while the dependence power indicates the number of factors that affect it.

Level partitions

The data in the Table 3 presents the final reachability matrix, which can be categorized into different levels. The reachability set and antecedent set for each factor were obtained from this matrix. The reachability set includes the factor itself and any other factors that it may assist in achieving, while the antecedent set comprises the factor itself and any other factors that may help in achieving it. The intersection set of the reachability set and antecedent set is then computed for all factors. Factors that have identical reachability and intersection sets are designated as level I and are positioned at the top of the ISM hierarchy. These top-level factors would not aid in attaining any other element beyond their own level, signifying that they would not contribute to mitigating any other factor above their level. The initial iteration of level partitioning results in level I, as demonstrated in Table 4. These levels aid in the construction of the diagraph and the final ISM model.

Formation of ISM

The data in the Tables 4 contain the level partition tables used to generate the structural model of the different critical factors required. Following the removal of transitivity using the ISM methodology, the diagraph was ultimately transformed into the Interpretive Structural Model, as demonstrated in Figure 4.

MICMAC analysis

The MICMAC (Matrice d'Impacts Croisés Multiplication Appliquée á un Classement or Cross-Impact Matrix Multiplication Applied to Classification) analysis is employed to identify and evaluate the driving power and dependence of the factors that contribute to the problem or ultimate goal. According to the MICMAC principle, which is based on matrix multiplication properties, if element X has a direct influence on element Y, and Y has a direct impact on element Z, then any changes to X can potentially affect Z.

Table 4. Level Partitioning

FN	Reachability Set	Antecedent Set	Level
F1	1,2,3,4,5,6,7,8,9,10,11,12,13,	1,2,3,6,8,10,11,12,13,	IV
F2	1,2,3,4,5,6,7,8,9,10,11,12,13,	1,2,3,5,6,10,12,13,	IV
F3	1,2,3,4,5,6,7,8,9,10,11,12,13,	1,2,3,4,6,8,10,11,12,13,	IV
F4	3,4,5,6,7,8,9,10,11,13,	1,2,3,4,5,6,8,10,11,12,13,	III
F5	2,4,5,6,7,8,9,10,11,13,	1,2,3,4,5,6,8,10,11,12,13,	III
F6	1,2,3,4,5,6,7,8,10,11,12,13,	1,2,3,4,5,6,8,10,11,12,13,	III
F7	7,8,13,	1,2,3,4,5,6,7,8,9,11,12,13,	I
F8	1,3,4,5,6,7,8,9,13,	1,2,3,4,5,6,7,8,9,10,11,12,13,	I
F9	7,8,9,13,	1,2,3,4,5,8,9,10,11,12,13,	II
F10	1,2,3,4,5,6,8,9,10,11,12,13,	1,2,3,4,5,6,10,11,12,13,	III
F11	1,3,4,5,6,7,8,9,10,11,12,13,	1,2,3,4,5,6,10,11,12,13,	III
F12	1,2,3,4,5,6,7,8,9,10,11,12,13,	1,2,3,6,10,11,12,13,	IV
F13	1,2,3,4,5,6,7,8,9,10,11,12,13,	1,2,3,4,5,6,7,8,9,10,11,12,13,	I

Table 5. MICMAC ranking of the factors

FN	Dependence power	Driving Power	Driving/Dependence power	MICMAC Rank
F1	9	13	1.44	3
F2	8	13	1.63	1
F3	10	13	1.30	4
F4	11	10	0.91	9
F5	11	10	0.91	9
F6	11	12	1.09	7
F7	12	3	0.25	13
F8	13	9	0.69	11
F9	11	4	0.36	12
F10	10	12	1.20	5
F11	10	12	1.20	5
F12	8	13	1.63	1
F13	13	13	1.00	8

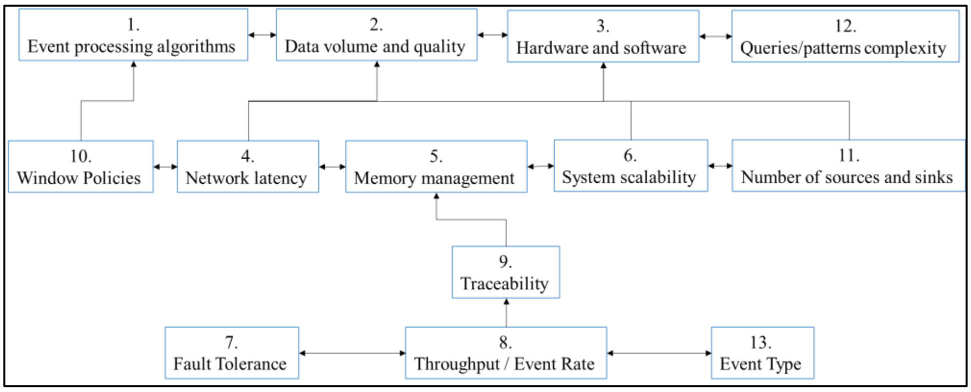

Figure 4. ISM Model

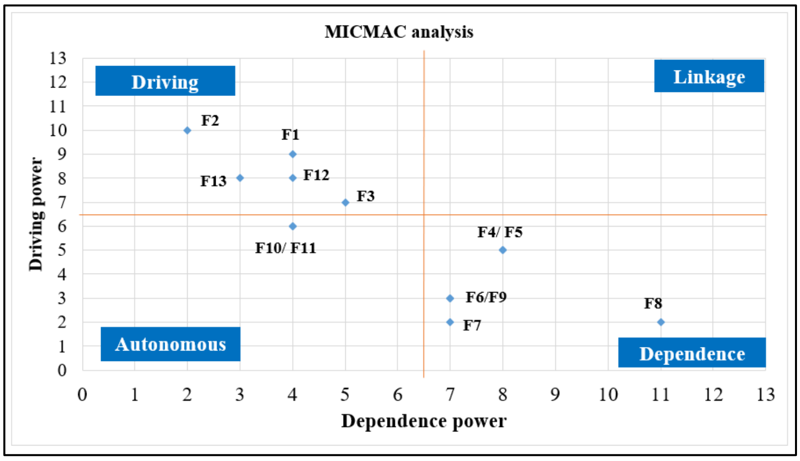

Figure 5. MICMAC analysis

This study utilizes the MICMAC analysis to examine the driving and dependent power of specific factors in the context of event based systems. The data in the Table 5 contains the driving power and dependence for each of these critical success factors, and subsequently, the driver power-dependence diagram is created, as illustrated in Figure 5.

Results and Discussion

The ISM technique is comprehensible to a diverse set of users across interdisciplinary teams. It offers a means of unifying varying viewpoints among the participating groups, and is capable of handling the many components and relationships inherent in complex systems. Its heuristic evaluation of model formulation adequacy yields valuable insights into system behaviour. Furthermore, the ISM method is user-friendly and available to a wider audience. Due to these characteristics, the ISM approach has gained widespread adoption.

When software developers and designers consider ways to improve their work, a multitude of factors come into play. Determining which factors to prioritize can be challenging due to the various constraints within the organization. To address this issue, the authors of this study utilized the ISM methodology and MICMAC analysis to examine the crucial aspects of event-based information systems performance. Their goal was to achieve effective and efficient performance improvement and design. After conducting a thorough literature review, they identified thirteen critical success factors and developed a hierarchical structure model using the ISM methodology. The study also explored the interdependence among the identified factors.

The data in the Table 3 shows the Reachability Matrix without transitivity, accounting for driving and dependence power. The factors with the highest driving power are ‘Event processing algorithms’, ‘Data volume and quality’, ‘Hardware and software’, ‘Queries/patterns complexity’, and ‘Event type’, followed by ‘System scalability’, ‘Window Policies’, ‘Number of sources and sinks’, among others. Factors with greater driving power have a stronger influence on other system factors. Improving driving factors can positively impact the performance of other factors, ultimately enhancing overall system performance. Managing factors such as fault tolerance, throughput, and event type can improve traceability, thereby strengthening network latency, memory management, system scalability, and other factors. On the other hand, factors with high dependence power, such as 'Throughput/Event Rate', have a stronger impact on other factors. Changes or improvements to these factors may have little effect on the other factors.

The data in the Figure 5 illustrates that ‘Window Policies’ and ‘Number of sources and sinks’ are located in Cluster I, which includes autonomous factors with low driving and dependence power. These factors can be considered separate and are heavily influenced by other factors. Meanwhile, Cluster II comprises critical success factors such as ‘Network latency’, ‘Memory management’, ‘System scalability’, ‘Fault Tolerance’, and ‘Throughput/Event Rate’, and ‘Traceability’ which are dependent factors. None of the critical success factors fall within Cluster III, which consists of linkage factors with both strong driving and dependence power. These factors are unstable in that any changes to them will affect other factors and result in a feedback effect on themselves. The absence of any factors within the linkage variable suggests that none of the critical success factors are inherently unstable. Remaining all the factors like ‘Event processing algorithms’, ‘Data volume and quality’, ‘Hardware and software’, ‘Queries/ patterns complexity’, and ‘Event type’ are the critical success factors positioned in cluster IV (Driving factors). The critical success factors substantial driving power and limited dependence necessitate treating them as vital to success. To achieve a substantial ergonomic improvement in the current system, policymakers and software developers should prioritize addressing these critical success factors.

The thirteen factors were ranked using a simple formula that divides a factor’s driving power by its dependence power. A higher ranking is achieved with more driving power and less dependence. Table 6 displays the rankings after the MICMAC analysis.

Table 6. Cluster formation as per IRM

Cluster No	Clusters	Factors
I	Autonomous	10,11
II	Dependence	4,5,6,7,8,9
III	Linkage	-
IV	Driving	1,2,3,12,13

Conclusion

This study utilized an extensive literature review and consultation with experts to determine the influencing factors and formulate an interpretive structural model of thirteen critical success factors (CSFs) for evaluating event-based systems. The proposed model effectively displays the interrelationships among these CSFs and can aid decision makers in understanding their relative importance for effective software development. Based on the model's hierarchy diagram (Figure 4), the ‘Data volume and quality’ CSF is the most important as it drives other CSFs and aids in strategic planning. It is recommended that policy makers and software management authorities prioritize policies and organizational structure to improve software performance. The CSFs such as fault tolerance, event type, and throughput occupy the lowest hierarchical level and represent desired goals. To achieve these goals, continuous improvement of the highest-level CSFs is necessary, which can be achieved through critical examination and alternative solution generation.

References

Boer, A., Winkels, R., van Engers, T., and de Maat, E. (2004). A content management system based on an event-based model of version management information in legislation. Legal Knowledge and Information Systems, 19-28.

Ciociola, A., Giordano, D., Vassio, L., and Mellia, M. (2023). Data driven scalability and profitability analysis in free floating electric car sharing systems. Information Sciences, 621:545-561. https://doi.org/10.1016/j.ins.2022.11.116

Cleland-Huang, J., Chang, C.K., and Christensen, M. (2003). Event-based traceability for managing evolutionary change. IEEE Transactions on Software Engineering, 29(9):796-810. https://doi.org/10.1109/TSE.2003.1232285

Cugola, G., Di Nitto, E., and Fuggetta, A. (2001). The JEDI event-based infrastructure and its application to the development of the OPSS WFMS. IEEE Transactions on Software Engineering, 27(9):827-850. https://doi.org/10.1109/32.950318.

Cugola, G., Di Nitto, E., and Fuggetta, A. (1998). Exploiting an event-based infrastructure to develop complex distributed systems. In Proceedings of the 20th International Conference on Software Engineering, 261-270. https://doi.org/10.1109/ICSE.1998.671135.

Fisher, O.J., Watson, N.J., Porcu, L., Bacon, D., Rigley, M., and Gomes, R.L. (2022). Data-driven modeling for resource recovery: Data volume, variability, and visualisation for an industrial bioprocess. Biochemical Engineering Journal, 185:108499. https://doi.org/10.1016/j.bej.2022.108499

Flouris, I., Giatrakos, N., Deligiannakis, A., Garofalakis, M., Kamp, M., and Mock, M. (2017). Issues in complex event processing: Status and prospects in the Big Data era. Journal of Systems and Software, 127:217-236. https://doi.org/10.1016/j.jss.2016.06.011

Hinze, A., Sachs, K., and Buchmann, A. (2009). Event-based applications and enabling technologies. Proceedings of the Third ACM International Conference on Distributed Event-Based Systems - DEBS '09. doi:10.1145/1619258.1619260

Joshi, M.P., and Deshpande, V. (2022). Application of interpretive structural modeling (ISM) for developing ergonomic workstation improvement framework. Theoretical Issues in Ergonomics Science, 24(2), 1-23. https://doi.org/10.1080/1463922X.2022.2044932

Liu, J.C., Hsu, C.H., Zhang, J.H., Kristiani, E., and Yang, C.T. (2023). An event-based data processing system using Kafka container cluster on Kubernetes environment - Neural Computing and Applications. SpringerLink. https://doi.org/10.1007/s00521-023-08326-1

Long, T., and Jia, Q.-S. (2023). On multi-scale event-based optimization. Results in Control and Optimization, 10. 100185. https://doi.org/10.1016/j.rico.2022.100185

Li, W., Sader, M., Zhu, Z., Liu, Z., and Chen, Z. (2023). Event-triggered fault-tolerant secure containment control of multi-agent systems through impulsive scheme. Information Sciences, 622, 1128-1140. https://doi.org/10.1016/j.ins.2022.11.132

Liu, L., Song, R., and Xia, L. (2023). Constrained event-driven policy iteration design for nonlinear discrete time systems. Neurocomputing, 528, 226-236. https://doi.org/10.1016/j.neucom.2023.01.060.

Mendes, M.R.N., Bizarro, P., and Marques, P. (2009). A Performance Study of Event Processing Systems. In: Nambiar, R., Poess, M. (eds) Performance Evaluation and Benchmarking. TPCTC 2009. Lecture Notes in Computer Science, vol 5895. https://doi.org/10.1007/978-3-642-10424-4_16

Prasadana, J.P. (2022). Compute-Based Analysis in Decision-Making Process. Journal of Information Systems and Management (JISMA), 2(2):17-24. https://doi.org/10.4444/jisma.v2i2.280

Ren, H., Zong, G., Qian, X., Yue, W., and Shi, K. (2023). Hybrid event-based asynchronous finite-time control for cyber-physical switched systems under denial-of-service attacks. Journal of the Franklin Institute, 360(2):1,036-1,057. https://doi.org/10.1016/j.jfranklin.2022.11.028

Warfield, J. (1974). Developing interconnection matrices in structural modeling. IEEE Transactions on Systems, Man, and Cybernetics, SMC-4(1), 81-87. https://doi.org/10.1109/TSMC.1974.5408524.

Yang, S., Tan, J., Lei, T., and Linares-Barranco, B. (2023). Smart Traffic Navigation System for Fault-Tolerant Edge Computing of Internet of Vehicle in Intelligent Transportation Gateway. IEEE Transactions on Intelligent Transportation Systems