Proceedings of the Design Society, Volume 6: DESIGN 2026
https://doi.org/10.1017/pds.2026.10631

Navigating the landscape of MBSE methodologies: a comparative study of ISO 15288 technical process coverage

Landry Bessault✉, Julia Guérineau and Louis Rivest

École de technologie supérieure, Canada

✉ landry.bessault.1@ens.etsmtl.ca

ABSTRACT: This paper compares nine Model-Based Systems Engineering methodologies with ISO 15288:2023 on the basis of their technical process coverage, to assess the extent to which the methodologies and standard are aligned. The technical processes covered include the definition of stakeholders, stakeholder needs, system requirements, architecture, design, verification, and validation. Two comparison criteria are also linked to the standard, namely, traceability and customisation. The results indicate that the definition of system requirements and system architecture are common core components of SE.

KEYWORDS: model-based systems engineering (MBSE), systems engineering (SE), design methods, product development, system modelling

1. Introduction

Systems Engineering (SE) is an interdisciplinary approach for the design, realisation and management of systems (Sillitto et al., 2019). The International Council on Systems Engineering (INCOSE) is one of the main organisations contributing to the domain (Sprague, 2025), and it co-developed, with the International Organisation for Standardisation (ISO), standard ISO 15288:2023. This standard is a critical document in the SE community (Dori, 2024; Panarotto et al., 2025). It is broken down into multiple technical processes that cover the system lifecycle and system development. The INCOSE Handbook (INCOSE, 2023) is aligned with the standard ISO 15288:2023 and forms a key industry document (Hamelin et al., 2010). Both the handbook and the standard take a high-level approach to system development. The Object Management Group (OMG) also operates in the SE domain and can be considered an authoritative source (Dahmann & Roedler, 2016). While INCOSE is more theory-oriented, OMG is more focused on developing and maintaining operational standards such as the System Modelling Language (SysML).

Within the broader discipline of SE, Model-Based Systems Engineering (MBSE), which has been greatly influenced by OMG and INCOSE (Friedenthal, 2007), operationalises the concepts of system development, as can be seen in the definition of MBSE. It is "*the formalised application of modelling to support system requirements, design analysis, verification and validation activities from the conceptual stage through the entire system lifecycle*" (INCOSE, 2007). MBSE builds upon SE principles by using model-centric processes in place of document-centric approaches (Parrott & Spayd, 2020). It makes it possible to have a single authoritative source of truth for systems information, which improves consistency, traceability and collaboration in multidisciplinary teams (Cohen et al., 2022; Guérineau et al., 2025; Subarna et al., 2020), and grasp the increasing complexity of systems (Hill et al., 2024). MBSE can be broken down into three pillars, namely modelling languages, modelling tools, and methodologies. This study focuses on MBSE methodologies, which are defined as "*a collection of related processes, methods and tools*" (Estefan, 2008). However, as reliance on these methodologies has grown, so has their diversity, which has led to the landscape being dense and difficult to navigate

https://doi.org/10.1017/pds.2026.10631 Published online by Cambridge University Press

(Ma et al., 2022). This fragmentation is echoed in the system development domain, where many existing contributions focus on methodologies, frameworks and techniques. For practitioners and researchers alike, the fragmentation often creates inflated claims (Campo et al., 2023) and more confusion than clarity (Call et al., 2024). In this study, ISO 15288:2023 technical processes are used as the basis of comparison due to the standard's importance in systems development. The research question addressed is: *how do MBSE methodologies, which are referenced by OMG, align with and support ISO 15288:2023 on the basis of their technical processes?*

In the sections that follow, this paper looks at the state of art, explains the comparison approach used, presents the results obtained and discusses them, and ends with a conclusion.

2. State of the art

There exist several studies that compare MBSE methodologies. However, to the best of the authors' knowledge, none compare them directly with the standard system development processes set out in ISO 15288:2023.

Some authors propose new methodologies to fill a gap in the landscape and perform widescale literature reviews of existing methodologies when doing so. This is the case for Diatte et al. (2022) and Mažeika (2021). The first puts forward a methodology to integrate system Reliability, Availability and Maintainability, and the second proposes a methodology for secure system design. Those authors' works do not explicitly compare methodologies but do list and explain a selection of them. In the same vein, Abdelrazik et al. (2019) compare a subset of methodologies in the domain of special-purpose machinery (SPM) to propose a new methodology that would better guide the user through the system development processes. The authors use a table to position the methodologies and custom system development processes that are not based strictly on ISO 15288:2023. They use a binary notation system for their comparison, which may not capture partial coverage.

While the above-mentioned works use literature reviews to justify new methodologies, other studies instead focus on comparison. For instance, Wilke et al. (2024) compare different MBSE methodologies and modelling languages using multiple criteria. Some of the criteria are linked to the ease of learning a methodology, a methodology's scalability, and other features that are unrelated to the system development processes covered in ISO 15288:2023. Both Abdelrazik et al. (2019) and Wilke et al. (2024) restrict the scope of their comparisons to SPM, but the methodologies they compare could be applied in other domains. Similarly, a two-part study focuses on comparing MBSE methodologies, with the custom comparative framework defined in Weilkiens et al. (2016) and applied in Di Maio et al. (2021). The framework for the evaluation of MBSE methodologies for practitioners includes modelling language, scalability, tailoring, and integration with other tools. The resulting comparison is thorough and does consider the standard, but it does not seem to detail the extent to which the methodologies compared cover the technical processes set out in the standard.

These recent comparative works build upon an earlier survey by Estefan (2008). To the authors' knowledge, that was the first work to consider a collection of methodologies and describe each one. A list of methodologies was prepared and later expanded upon by INCOSE before being made available by OMG (2018). It constitutes a starting point for studying methodologies. The document, however, does not aim to compare the different methodologies, but rather describes their steps, and their corresponding tools (Estefan, 2008; OMG, 2018).

Overall, these papers either compare a limited selection of methodologies with the standard ISO 15288:2023 or use comparison tables that are not explicitly rooted in system development processes from the standard. Furthermore, prior work tends to emphasise methodological characteristics such as tool integration or learnability rather than measuring the methodologies against the standard. As a result, the extent to which contemporary MBSE methodologies align with, or support, the lifecycle processes prescribed by ISO 15288:2023 remains insufficiently explored despite the standard's prominence (Dori, 2024). In line with these observations, this study addresses the literature gap by comparing methodologies directly with the standard's system development processes.

3. Research approach

The research approach used involves systematically identifying and analysing multiple MBSE methodologies and is explained in four subsections. It makes it possible to cover the standard ISO 15288:2023 as part of the research question and select methodologies for comparison. The first

https://doi.org/10.1017/pds.2026.10631 Published online by Cambridge University Press

subsection defines the comparison grid, which is based on ISO 15288:2023. The second explains the reason for choosing a three-point grading scale linked to the definition of the comparison grid. The third subsection presents how the methodologies were selected, and the fourth pertains to the identification of the documentation used for each methodology.

3.1. The comparison grid

The comparison grid includes the system lifecycle technical processes from ISO 15288:2023 that are directly related to system development, plus Customisation and Traceability. The technical processes set out in ISO 15288:2023 form the basis of the INCOSE Handbook (INCOSE, 2023) and are presented in Figure 1. The processes that are directly related to system development and therefore relevant to the comparison are indicated in blue in Figure 1, while those in grey are not and therefore outside of the scope of this paper. For instance, the Maintained System process belongs to the system operation phase rather than the system design phase and is therefore not relevant to the focus of this paper. The Validation steps (in orange) and Verification steps (in white) are performed after each technical process. Validation & Verification (V&V) are grouped together in this study for easier interpretation of the results. The product development processes and V&V form the basis of the comparison criteria. The paragraphs that follow describe the relevant processes in detail.

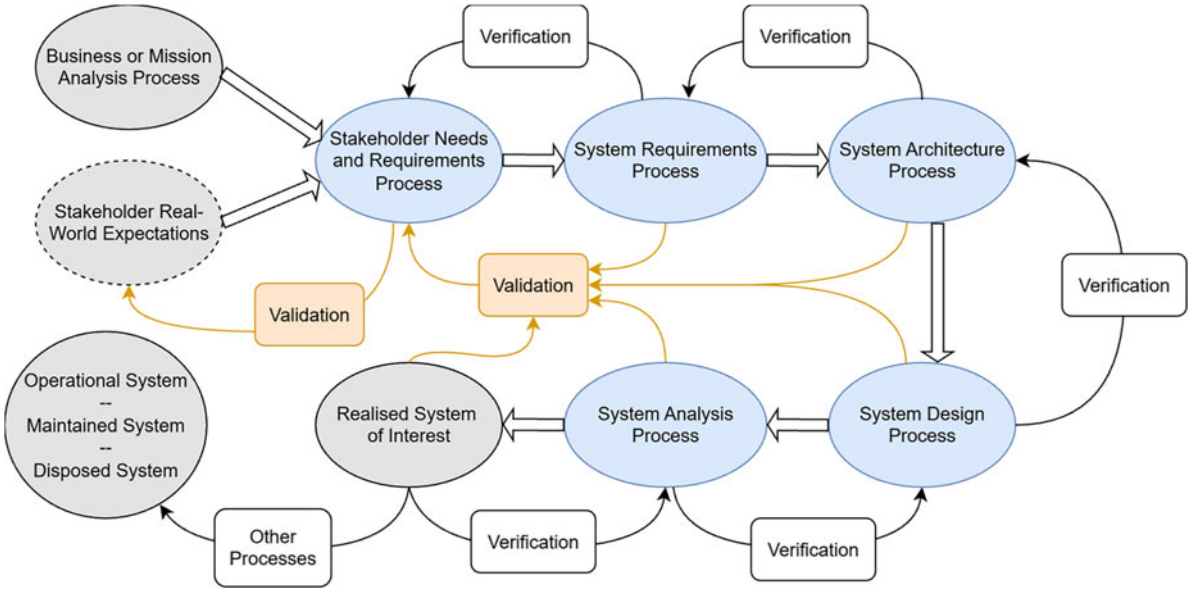


Figure 1. The technical processes set out in ISO 15288:2023 adapted from INCOSE (2023)

Firstly, the purpose of the Stakeholder Needs and Requirements process is to list the various stakeholders "*involved with the System [of Interest] throughout its lifecycle*" and their needs (ISO 15288:2023). This process is split into two subprocesses—Stakeholder Identification and Stakeholder Needs—in line with its definition in the standard. These subprocesses represent two of the criteria in the comparison grid. These subprocess names differ from the name shown in Figure 1 and do not include the word "requirement" for consistency, to keep the term "needs" reserved for stakeholders and the term "requirements" used strictly for the characteristics the system must meet and to avoid confusion (Katz et al., 2024). The goal of the System Requirements process, another comparison grid criterion, is to transform the needs identified into measurable technical characteristics that the system of interest should meet. The System Architecture process, also a criterion, produces a principle-based solution that satisfies the requirements of the system while trying to avoid including specifics about the design or the making of the system. The System Design process, yet another criterion, covers the development of the system of interest's blueprint with the objective of detailing it enough to enable its making. For this study, the goal of the System Analysis process is to evaluate and compare candidate system solutions based on their ability to meet the system requirements. This process emphasises candidate selection from the ISO 15288:2023 definition, as it is the most tangible part of the definition and is used as a criterion in the comparison grid. Likewise, the V&V processes, which serve as another

https://doi.org/10.1017/pds.2026.10631 Published online by Cambridge University Press

criterion, are grouped together as they both span the entirety of system development and address a similar objective—to assess whether the outputs of a process align with the previous results or Stakeholder Real-World Expectations.

Two additional comparison criteria—Customisation and Traceability—are also included in the comparison grid. Customisation tailors an entire methodology and enables a user to modify it based on the needs of the system being developed, as mentioned in ISO 15288:2023. This is one aspect that is needed for smoother adoption of MBSE (Ammersdörfer et al., 2023). Traceability, as it is defined by Katz et al. (2024) in the Systems Engineering Book of Knowledge (SEBoK) and used in the standard, means "*[t]he degree to which a relationship can be established between two or more systems of the development process, especially systems having a predecessor-successor or master-subordinate relationship to one another*". It is specified for each technical process in ISO 15288:2023 and reported as a single criterion in this comparison. The next subsection outlines the grading scale chosen to evaluate each methodology's alignment with the standard.

3.2. The grading scale used to compare the methodologies

The MBSE methodologies are graded using a three-point scale for its ease of use and understanding (Preston & Colman, 2000) and as it aligns with the objective of improving clarity in the SE domain. The grading scale represents the coverage of the criteria, with three options: none, partial and full. In other words, a methodology has an empty dot in the comparison grid if the criterion in question is neither explicitly nor implicitly mentioned in the documentation considered for said methodology. If a criterion is partially or implicitly covered in the documentation, a half-dot grade is given. Finally, the methodology has a full dot if the criterion in question is fully covered and explicitly explained in the documentation. Detailed justifications for the grades given to the methodologies are provided in the results section.

3.3. The selection of the methodologies

The methodologies considered were extracted from the OMG (2018) website and Wilke et al. (2024), which list a total of twenty-two unique methodologies. To ensure the methodologies were understood consistently and in adequate depth, priority was given to comprehensive documentary sources rather than fragmented documents from different sources, and only one document was selected per methodology. A MBSE methodology defined under a single name is expected to have a comprehensive document defining it. Most of the documents selected are reference books or papers written by the methodology creator or an authoritative author. Furthermore, only publicly available documents were selected and the authors ensured to choose the latest version. For instance, the book chosen for the ARCADIA methodology, developed by Thales, was written by a former Thales employee and the paper studied for the CONSENS methodology was a founding document. However, in the case of Alstom's ASAP methodology, for instance, no comprehensive document was found, so it is not included in the comparison. Furthermore, the document for each methodology should have a coherent structure, meaning that if it omitted any steps in the product development process, they should be easily identifiable. For instance, for the MagicGrid methodology, its document clearly states that the System Design process is outside of its scope. This would ensure that the documentation indicates its own limitations and the MagicGrid book is therefore representative of the coverage methodology. This operation was applied to all the methodologies. In addition, Harmony agile MBSE (aMBSE) was determined to be an evolution of IBM's Rational Unified Process for Systems Engineering (RUP-SE) methodology and the Rational Software Corporation's Rational Unified Process (RUP) methodology. Indeed, IBM's web page presenting RUP-SE was deleted around 2015 and the use of RUP was confirmed in white paper obtained through the Internet Archive (Rational Software Corporation, 2011). This suggests that RUP-SE is no longer IBM's prime solution and has been superseded by Harmony aMBSE. RUP-SE and RUP are therefore not included in the comparison. Only nine of the original twenty-two methodologies were found to satisfy all our selection criteria. They are outlined in the next subsection.

3.4. The identification of corresponding documentation

The documents studied for OOSEM (Friedenthal et al., 2015) and SysMOD (Weilkiens et al., 2022) are both reference books written by SysML founders. Some of the methodologies compared were proposed

https://doi.org/10.1017/pds.2026.10631 Published online by Cambridge University Press

by companies: Thales' ARCADIA (Voirin, 2018), IBM's Harmony aMBSE (Douglass & IBM Corporation, 2017) and Dassault Systèmes' MagicGrid (Aleksandravičienė & Morkevičius, 2021). Others were proposed by researchers, such as OPM (Dori, 2016), SPES_XT (Pohl et al., 2016) in a book, and LITHE (Ramos et al., 2013) and CONSENS (Gausemeier et al., 2013) in research papers. Note that the SPES_XT methodology was chosen over the SPES methodology as the XT stands for "extended" and means that SPES_XT includes SPES.

These methodologies emerged at different times and can be represented on a timeline. Figure 2 shows the date of publication of the methodologies. The regular (non-bold) methodology names represent the beginnings of the methodologies in question. These instances are often reported in a short paper or a press release from the company. In contrast, the names in bold represent the year of publication of a comprehensive document, according to the authors' knowledge. Note that a comprehensive document such as a book could be the first publication of a methodology, which is the case for OPM where Dori released a book (Dori, 2016). For ARCADIA, for instance, Thales developed it and released it in 2015, while a book of knowledge was published in 2017. For LITHE and CONSENS, as only research papers were found, only their publication is shown in the timeline. Furthermore, for SPES and SPES_XT, both are presented in the timeline as SPES_XT is an extension of the SPES methodology, therefore the methodology found its roots mostly in SPES. The next section presents the results of the analysis of the nine methodologies using the comparison grid.

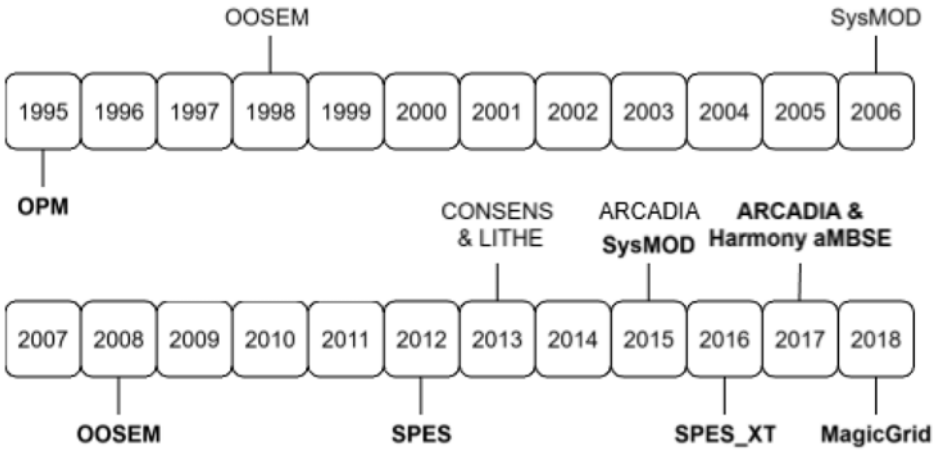


Figure 2. Timeline of the release of different MBSE methodologies

4. Results

This section presents the comparison results for the nine MBSE methodologies considered (see Table 1). Each methodology is evaluated in accordance with the nine criteria presented above. The Customisation and Traceability criteria are presented before those for the technical processes from ISO 15288:2023. The methodologies are presented in descending order of criteria coverage.

It can be observed from Table 1 that Customisation is explicitly covered by only two methodologies: SysMOD and ARCADIA. For instance, SysMOD's documentation states that after the methodology has been tailored, employees should be trained for the specific practices included in the tailored methodology. SysMOD and ARCADIA therefore have full dots for this criterion, while the other seven methodologies have empty dots. For the Traceability criterion, the link between each system development phase is evaluated. The documentation for OOSEM, SysMOD, ARCADIA and MagicGrid explicitly mentions a link between each and every phase of system development, so these methodologies were each given a full dot for this criterion. Likewise, the documentation for SPES_XT explicitly links its artefacts, and details how the requirements are linked to the architecture definition. Since SPES_XT satisfies the Traceability criterion across all the processes it covers, it was also given a full dot. Harmony aMBSE has a half-dot because its documentation mentions a connection between Stakeholder Needs and System Requirements but does not explicitly mention any other links. In contrast, the documentation for OPM, CONSENS and LITHE does not

https://doi.org/10.1017/pds.2026.10631 Published online by Cambridge University Press

explicitly mention Traceability between processes. OPM's model is uninterrupted, with elements continuously linked and added to it. However, the concept of master-subordinate from the SeBOK definition is not explicitly mentioned in this methodology's documentation, therefore, it was given a half dot for Traceability. For LITHE and CONSENS, the documentation provided no detailed explanation of links between system development technical processes, so these methodologies have empty dots for Traceability.

The remaining rows of Table 1 pertain to system development technical processes. The documentation for ARCADIA, OOSEM and SysMOD addresses all the processes considered, with ARCADIA and OOSEM only partially covering Stakeholder Identification as this process is implicitly, but not explicitly, mentioned. Since a stakeholder listing is implied to form Stakeholder Needs, this means that the stakeholders have to be listed. This implicit identification is considered to provide partial coverage, as explained earlier.

Harmony aMBSE has a full dot grade for the Stakeholder Identification criterion, thanks to its preliminary step, whereas MagicGrid's grade is a half-dot as its documentation implies the need for a stakeholder list. MagicGrid's document details the gathering of Stakeholder Needs and the fact that it leads to the definition of System Requirements, while Harmony aMBSE begins with a project initialisation which seeks stakeholder use cases. In both methodologies, this process is followed by System Architecture. The Harmony aMBSE document clearly states that there should be a transition from the systems engineers to specialised engineers for System Design. The MagicGrid document, for its part, builds the System Architecture by detailing every subsystem, but does not expressly explain the System Design. It requires a transition to specialised engineers for System Design but does not provide details about this transition. Neither methodology includes candidate solution comparison; however, the Harmony aMBSE methodology does include a System Architecture compromise analysis step that is not present in other methodologies or the comparison grid.

For the CONSENS and LITHE methodologies, both of their documents begin with the definition of System Requirements without specifying its inputs. They also cover the System Architecture. The CONSENS methodology's partial models make it possible to design the shape of the system, which covers System Design. Similarly, the LITHE methodology is used to develop hardware and software units. These methodologies diverge when it comes to candidate comparison though, with the LITHE documentation incorporating the comparison of candidate system solutions in architecture definition while the CONSENS documentation does not explicitly mention this step. Similarly, there are discrepancies in these methodologies' approaches to V&V. CONSENS uses multiple partial models to verify that the system is built in accordance with the other partial models—this is implicit in the explanation of the methodology—whereas LITHE verifies the system during System Design and validates it right before deployment.

Lastly, the documentation for OPM and SPES_XT indicates the scope of these methodologies is narrower and both start with System Requirement followed by System Architecture. The concept is similar in SPES_XT and CONSENS, with multiple views forming a model that incorporates the requirements and the architecture definition. In the OPM documentation, the stakeholders are first implicitly listed. Then, the System Requirements and System Architecture are constructed using the custom modelling language Object-Process Language. This custom language simplifies simulation and makes it easier to understand the system's logic (Basnet et al., 2020). When it comes to System Design, OPM does not cover it while SPES_XT covers the design of the software but not the physical part of the system. As for V&V, the documentation for OPM does not explicitly mention it; however, the logical simulation of blocks and their visual link between one another facilitate the early verification once the model has been created, which is why the methodology was given a half dot for V&V, whereas the documentation for SPES_XT does not mention V&V implicitly or explicitly.

This study suggests there is variation in the coverage of technical processes among the MBSE methodologies analysed. ARCADIA, OOSEM, and SysMOD stand out as they cover all the technical processes considered, whereas other methodologies, like Harmony aMBSE, do not. These results are further discussed in the following section in light of prior studies with a view to highlighting key points, limitations and future perspectives.

https://doi.org/10.1017/pds.2026.10631 Published online by Cambridge University Press

Table 1. The comparison grid of 9 MBSE methodologies

Criteria \ Methodologies	SysMOD	ARCADIA	OOSEM	MagicGrid	Harmony aMBSE	CONSENS	LITHE	OPM	SPES_XT
Traceability	●	●	●	●	◒	○	○	◒	●
Customisation	●	●	○	○	○	○	○	○	○
Stakeholder Identification	●	◒	◒	◒	●	◒	●	◒	○
Stakeholder Needs	●	●	●	●	◒	○	○	○	○
System Requirements	●	●	●	●	●	●	●	●	●
System Architecture	●	●	●	●	●	●	●	●	●
System Design	●	●	●	○	○	●	●	○	◒
System Analysis	●	●	●	○	◒	○	●	○	○
Verification and Validation	●	●	●	○	◒	◒	●	◒	○

5. Discussion

The results obtained suggest that only three of the nine methodologies studied cover all the system development technical processes. They are aligned with the findings of Di Maio et al. (2021) for OPM and ARCADIA and those of Abdelrazik et al. (2019) for Harmony SE and OOSEM. The findings in Wilke et al. (2024) and Table 1 are similar apart from the fact that the paper considers a "physical structure" criterion in place of the comparison grid's System Design criterion and their results differ. It is therefore difficult to further investigate the latter criterion as it is not explicitly defined in the paper.

In comparing the user guidance provided in the documentation considered for the methodologies, SysMOD stands out for its explanation of the different steps. The documents or models required to begin each phase, the personnel involved, and the artefacts to be produced by the end of each phase are clearly specified. Likewise, for MagicGrid, ARCADIA and OOSEM tangible guidance is provided that includes examples and almost tutorial-like instructions. The same cannot be said for CONSENS and LITHE, whose documentation remains theoretical, which can be explained by the fact they are articles and therefore shorter in length. MagicGrid, LITHE, OPM, and SPES_XT take a recursive approach, with detailed and repeated system decomposition. In contrast, the ARCADIA, and Harmony aMBSE methodologies follow progress through the development steps in a more linear fashion.

In terms of relevance for academics, the results show that all the methodologies cover the definition of System Requirements and System Architecture, which suggests that these processes constitute the common core components of MBSE as proposed by Hermelingmeier et al. (2025). Nonetheless, the fact that there are few common core components indicates there may be a disconnect between INCOSE's vision of implementing SE across the system lifecycle and the extent of SE implementation in current MBSE methodologies. Furthermore, the results suggest that the older methodologies, with the exception of OPM, tend to have better coverage than those released more recently. This paper could serve as a starting point for further comparison and possibly the improvement of existing MBSE methodologies.

In terms of relevance for the industry, MBSE methodologies are an important part of MBSE implementation (Henderson et al., 2024). They form the link between the higher-level guidelines of SE and the more operational means of working with MBSE. For instance, MBSE methodologies make it

https://doi.org/10.1017/pds.2026.10631 Published online by Cambridge University Press

possible to bridge technical processes and the concrete use of tools, like the languages chosen. INCOSE recognises multiple methodologies (OMG, 2018) but does not recommend any in particular, which adds a layer of difficulty when adopting MBSE (Call et al., 2024). It should be noted that the methodologies are a major determining factor in the success of MBSE adoption. This assertion is corroborated by the results of an MBSE survey conducted by INCOSE in 2020, which indicated that the methodologies constitute the primary obstacle to MBSE adoption (INCOSE, 2020). This work complements Weilkiens et al. (2016) and Wilke et al. (2024) by providing an overview of different methodologies and comparing their coverage of system development technical processes and their corresponding criteria.

In a wider context, methodologies are not the sole pillar of MBSE to consider—there are also tools and modelling languages—and the choice made for one pillar may limit one's options for the others. For instance, the Harmony aMBSE methodology requires use of the tool IBM Rhapsody and the modelling language SysML v1. Likewise, in choosing ARCADIA, it is necessary to use the corresponding software, Capella, which uses its own modelling language. When it comes to modelling language, using the newer SysML v2 would surely change the artefacts made during the system development technical processes (Friedenthal, 2025). However, the methodologies' coverage of the technical processes would likely change very little. Nonetheless, all of the methodologies assessed are still based on SysML v1. The transition to SysML v2 may take a few years, but SysML v1 will nevertheless continue to be used to maintain companies' legacy models, which will ensure the continued relevance of this paper.

In the interest of ensuring this comparison is fair and replicable, this study intentionally considers methodology documentation from authoritative sources. This approach ensures a coherent analysis of the methodologies' alignment with the standard although it may not capture the latest updates to each methodology. For instance, for the MagicGrid methodology, there exists a standalone article detailing V&V (Morkevičius et al., 2022), but it does not contribute to the comparison grid since it is not included in the authoritative book considered for said methodology. Furthermore, this study is literature-based, and the text analysis is open to interpretation. Nonetheless, to limit subjectivity and ensure consistency, the authors use explicit definitions for the processes and criteria. The authors also acknowledge there may be discrepancies between how a methodology is described in the literature and how it is experienced by MBSE practitioners. This literature-based study could therefore be complemented by an empirical study with practitioners to deepen the understanding of this topic.

6. Conclusion

The paper compares nine MBSE methodologies with ISO 15288:2023 on the basis of their technical process coverage according to documentation published by authoritative sources – one document per methodology. The processes that all the methodologies and the standard have in common are System Requirement definition and System Architecture definition which seem to be the common core components of MBSE. This contrasts with INCOSE's vision for system development, which is to have Stakeholder Needs and System Design also be core components. Only three of the nine MBSE methodologies analysed are aligned with and support the system development standard ISO 15288:2023 according to the comparison grid. Those three methodologies are SysMOD, OOSEM and ARCADIA. This paper is a preliminary study that aims to facilitate the understanding of the MBSE methodologies landscape as seen through the lens of ISO 15288:2023.

References

Abdelrazik, M., Elsheikh, A., Zayan, M., & Elhady, A.-B. (2019). New Model-Based Systems Engineering Methodology Based on Transdisciplinary Quality System Development Lifecycle Model. *Journal Européen Des Systèmes Automatisés*, 52(5), 465–476. https://doi.org/10.18280/jesa.520505

Aleksandravičienė, A., & Morkevičius, A. (2021). *MagicGrid® Book of Knowledge Practical Guide to Systems Modeling using MagicGrid from Dassault Systèmes* (2nd edition). Dassault Systèmes.

Ammersdörfer, T., Inkermann, D., Müller, J., Mandel, C., Albers, A., Tekaat, J., Schierbaum, A., Anacker, H., Bitzer, M., Kleiner, S., Herrmann, J.-P., & Krause, P. (2023). Supporting Systems Engineering Activities By Artifact-Oriented Description And Selection Of Methods. *Proceedings of the Design Society*, 3, 3245–3254. https://doi.org/10.1017/pds.2023.325

Basnet, S., Valdez Banda, O. A., Chaal, M., Hirdaris, S., & Kujala, P. (2020). Comparison of system modelling techniques for autonomous ship systems. *Proceedings of the International Seminar on Safety and Security of Autonomous Vessels (ISSAV) and European STAMP Workshop and Conference (ESWC) 2019* (pp. 125–139).

https://doi.org/10.1017/pds.2026.10631 Published online by Cambridge University Press

Call, D. R., Herber, D. R., & Conrad, S. A. (2024). The Effects of the Assessed Perceptions of MBSE on Adoption. *INCOSE International Symposium*, 34(1) (pp. 462–478). https://doi.org/10.1002/iis2.13157
Campo, K. X., Teper, T., Eaton, C. E., Shipman, A. M., Bhatia, G., & Mesmer, B. (2023). Model-based systems engineering: Evaluating perceived value, metrics, and evidence through literature. *Systems Engineering*, 26(1) (pp. 104–129). https://doi.org/10.1002/sys.21644
Cohen, J., Kaetzer, M. S., Lumpkins, S., Rubin, D., & McGuire, K. (2022). A Model-Based Systems Engineering Journey to Developing a Concept of Operations (ConOps). *2022 IEEE Aerospace Conference (AERO)* (pp. 1–14). https://doi.org/10.1109/AERO53065.2022.9843691
Dahmann, J., & Roedler, G. (2016). Moving towards standardization for system of systems engineering. *2016 11th System of Systems Engineering Conference (SoSE)* (pp. 1–6). https://doi.org/10.1109/SYSOSE.2016.7542953
Di Maio, M., Weilkiens, T., Hussein, O., Aboushama, M., Javid, I., Beyerlein, S., & Grotsch, M. (2021). Evaluating MBSE Methodologies Using the FEMMP Framework. *2021 IEEE International Symposium on Systems Engineering (ISSE)* (pp. 1–8). https://doi.org/10.1109/ISSE51541.2021.9582465
Diatte, K., O'Halloran, B., & Van Bossuyt, D. L. (2022). The Integration of Reliability, Availability, and Maintainability into Model-Based Systems Engineering. *Systems*, 10(4), 101. https://doi.org/10.3390/systems10040101
Dori, D. (2016). *Model-Based Systems Engineering with OPM and SysML*. Springer New York. https://doi.org/10.1007/978-1-4939-3295-5
Dori, D. (2024). Model-based standards authoring: ISO 15288 as a case in point. *Systems Engineering*, 27(2), 302–314. https://doi.org/10.1002/sys.21721
Douglass, B. P. & IBM Corporation. (2017). Harmony aMBSE Deskbook Version 1.00 Agile Model-Based Systems Engineering Best Practices with IBM Rhapsody.
Estefan, J. A. (2008). Survey of Model-Based Systems Engineering (MBSE) Methodologies. INCOSE MBSE Initiative.
Friedenthal, S. (2007). Object Management Group Support for Model-Based Systems Engineering Standards. *INSIGHT*, 10(2), 10–12. https://doi.org/10.1002/inst.200710210
Friedenthal, S. (2025). The Next Generation Systems Modeling Language (SysML v2). *INCOSE*. https://www.incose.org/docs/default-source/content-library/the_next_generation_systems_modeling_language-sysml_v2-sfriedenthal-incose_sectoriii.pdfsfvrsn=813051c7_1
Friedenthal, S., Moore, A., & Steiner, R. (2015). *A practical guide to SysML: The systems modeling language* (3rd edition). Elsevier/Morgan Kaufmann.
Gausemeier, J., Gaukstern, T., & Tschirner, C. (2013). Systems Engineering Management Based on a Discipline-Spanning System Model. *Procedia Computer Science*, 16, 303–312. https://doi.org/10.1016/j.procs.2013.01.032
Guérineau, J., Rivest, L., & Bricogne, M. (2025). A synthesis mapping of approaches, processes, methods and tools from the scientific literature for multidisciplinary product development. *Results in Engineering*, 28, 108033. https://doi.org/10.1016/j.rineng.2025.108033
Hamelin, R. D., Walden, D. D., & Krueger, M. E. (2010). 4.4.2 INCOSE Systems Engineering Handbook v3.2: Improving the Process for SE Practitioners. *INCOSE International Symposium*, 20(1) (pp. 532–541). https://doi.org/10.1002/j.2334-5837.2010.tb01087.x
Henderson, K., McDermott, T., & Salado, A. (2024). MBSE adoption experiences in organizations: Lessons learned. *Systems Engineering*, 27(1), 214–239. https://doi.org/10.1002/sys.21717
Hermelingmeier, D., Bita, I. M., Menne, L., Jobelius, J., Tichonov, D., Pfeifer, S., & Dumitrescu, R. (2025). Maker Systems Engineering Systematic: A Practitioner-Driven Research Agenda. *Procedia CIRP*, 136, 844–849. https://doi.org/10.1016/j.procir.2025.08.144
Hill, T. R., Carnevale, A. E., Morris-Eckart, A., Sundaram, V., & Farhaj, L. Z. (2024). NASA's Use of MBSE and SysML Modeling to Architect the Future of Human Exploration. *INCOSE International Symposium*, 34(1) (pp. 527–543). https://doi.org/10.1002/iis2.13161
International Council on Systems Engineering (INCOSE). (2007). *Systems Engineering Vision 2020*. International Council on Systems Engineering (INCOSE).
International Council on Systems Engineering (INCOSE). (2020). Benchmarking the Benefits and Current Maturity of Model-Based Systems Engineering across the Enterprise. International Council on Systems Engineering (INCOSE). Systems engineering handbook: A guide for system life cycle processes and activities (5th edition). Wiley.
International Organisation for Standardisation (ISO). (2023). *Systems and software engineering — System life cycle processes (ISO 15288:2023)*. ISO.
Katz, T., Wheatcraft, L., & Ryan, M. (2024). *The Guide to the Systems Engineering Body of Knowledge (SEBoK) (2.12)*. N. Hutchison (Editor in Chief). The Trustees of the Stevens Institute of Technology.
Ma, J., Wang, G., Lu, J., Vangheluwe, H., Kiritsis, D., & Yan, Y. (2022). Systematic Literature Review of MBSE Tool-Chains. *Applied Sciences*, 12(7), 3431. https://doi.org/10.3390/app12073431
Mažeika, D. (2021). Model-Based Systems Engineering Method For Creating Secure Systems [Doctoral thesis, Kauno technologijos universitetas].

https://doi.org/10.1017/pds.2026.10631 Published online by Cambridge University Press

Morkevičius, A., Aleksandravičienė, A., & Strolia, Z. (2022). System Verification and Validation Approach Using the MagicGrid Framework. *INCOSE International Symposium*, 32(1) (pp. 767–781). https://doi.org/10.1002/iis2.12962

Object Management Group (OMG). (2018). Mbse: methodology [MBSE Wiki]. https://www.omgwiki.org/MBSE/doku.php?id=mbse:methodology

Panarotto, M., Isaksson, O., & Andersson, P. (2025). Value-driven MBSE for evolving business contexts: A quantitative study on aerospace electrification. *Design Science*, 11, e36. https://doi.org/10.1017/dsj.2025.10024

Parrott, E. L., & Spayd, L. C. (2020). Configuration and Data Management of the NASA Power and Propulsion Element MBSE Model(s). *2020 IEEE Aerospace Conference* (pp. 1–11). https://doi.org/10.1109/AERO47225.2020.9172375

Pohl, K., Broy, M., Daembkes, H., & Hönninger, H. (2016). *Advanced Model-Based Engineering of Embedded Systems*. Springer International Publishing. https://doi.org/10.1007/978-3-319-48003-9

Preston, C. C., & Colman, A. M. (2000). Optimal number of response categories in rating scales: Reliability, validity, discriminating power, and respondent preferences. *Acta Psychologica*, 104(1) (pp. 1–15). https://doi.org/10.1016/S0001-6918(99)00050-5

Ramos, A. L., Ferreira, J. V., & Barcelo, J. (2013). LITHE: An Agile Methodology for Human-Centric Model-Based Systems Engineering. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(3) (pp. 504–521). https://doi.org/10.1109/TSMCA.2012.2207888

Rational Software Corporation. (2011, August 1). Whitepapers—Rational Unified Process Whitepaper: Best Practices for Software Development Teams. https://web.archive.org/web/20110801031301/http://www.augustana.ab.ca/~mohrj/courses/2000.winter/csc220/papers/rup_best_practices/rup_bestpractices.html

Sillitto, H., Martin, J., McKinney, D., Griego, R., Dori, D., Krob, D., Godfrey, P., Arnold, E., & Jackson, S. (2019). Systems Engineering and Systems Definitions. INCOSE.

Sprague, C. (2025). The State of the Systems Engineering Discipline: A Longitudinal Analysis of INCOSE International Symposium Contributions (2012–2025). *INSIGHT*, 28(2) (pp. 9–48). https://doi.org/10.1002/inst.12534

Subarna, S., Jawale, A. K., Vidap, A. S., Sadachar, S. D., Fliginger, S., & Myla, S. (2020). Using a Model Based Systems Engineering Approach for Aerospace System Requirements Management. *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)* (pp.1–8). https://doi.org/10.1109/DASC50938.2020.9256589

Voirin, J.-L. (2018). *Model-based system and architecture engineering with the arcadia method*. ISTE Press.

Weilkiens, T., Lamm, J. G., Roth, S., & Walker, M. (2022). *Model-based system architecture* (2nd edition). Wiley. https://doi.org/10.1002/9781119746683

Weilkiens, T., Scheithauer, A., Di Maio, M., & Klusmann, N. (2016). Evaluating and comparing MBSE methodologies for practitioners. *2016 IEEE International Symposium on Systems Engineering (ISSE)* (pp. 1–8). https://doi.org/10.1109/SysEng.2016.7753174

Wilke, D., Humpert, L., Suwal, P., & Dumitrescu, R. (2024). Framework for the Selection of Systems Engineering Modelling Languages and Methods for Special Purpose Machinery. *Proceedings of NordDesign 2024* (pp. 549–557). https://doi.org/10.35199/NORDDESIGN2024.59

https://doi.org/10.1017/pds.2026.10631 Published online by Cambridge University Press