Proceedings of the Design Society, Volume 6: DESIGN 2026
https://doi.org/10.1017/pds.2026.10628

# An MBSE approach for integrating and computing LCA impact indicators within SysML models

**Onur Angin✉ and Detlef Gerhard**

*Ruhr University Bochum, Germany*

✉ onur.angin@ruhr-uni-bochum.de

**ABSTRACT:** Early design decisions are central to many circular economy approaches and largely determine environmental impacts. Yet LCA practice is often fragmented and difficult to integrate into engineering development processes. This paper proposes an MBSE approach that represents and computes LCA-based (LCIA) impact indicators within SysML models using reusable, configurable model elements and automated import of characterization factors. A proof-of-concept implementation demonstrates technical feasibility, illustrates process integration, and discusses limitations and future work.

**KEYWORDS:** life cycle assessment (LCA), model-based systems engineering (MBSE), SysML, life cycle impact assessment (LCIA), environmental impact assessment

## 1. Introduction

Early design choices strongly shape the implementation of circular economy approaches, therefore, assessing potential environmental impacts during product development is crucial. Around 80% of a product's total environmental impact is determined at the design stage (Diaz et al., 2021). **Life Cycle Assessment (LCA)** provides a structured method to quantify potential environmental impacts across the product life cycle and thereby inform design decisions. In practice, however, this often proves to be time-consuming, resource- and therefore cost-intensive (Lipšinić & Pavković, 2023; Teixeira, 2025). Existing assessment methods such as LCA are complex, difficult to access, and only partially transferable across industries, products, or system boundaries (Finkbeiner et al., 2014). The complexity of products and inconsistent data availability further challenge LCA application (Inkermann, 2022). Moreover, many widely used approaches take the form of normative guidance, such as ISO standards (ISO 14040; ISO 14044; ISO 14046; ISO 14067), which define principles, requirements, and reporting rules for LCA and related footprints, but do not prescribe a fully operational, machine-executable calculation procedure. As a result, assessments are often implemented through specialized toolchains and heterogeneous modeling choices, leading to fragmented applications that reduce comparability, hinder reproducibility, and weaken traceability of results. (Hunger et al., 2025; Popowicz et al., 2025; Rotondo et al., 2025). This highlights the need for a computable, standardized and traceable way to implement LCA impact-indicator calculations that can be integrated into engineering workflows.

While specialized LCA software tools such as *openLCA* (Ciroth, 2007) exist, they are typically applied in isolation and lack continuous integration into the product development process. Embedding environmental impact assessment within **Model-Based Systems Engineering (MBSE)** instead links evaluation directly to the system model, ensuring a consistent and traceable development workflow and enabling continuous, iterative assessment and validation as the design evolves.

At the same time, regulatory requirements and Ecodesign frameworks are increasing at both European and international levels, making robust environmental impact assessments indispensable (INCOSE, 2021). To address this efficiently, a *reusable, modular and domain-independent* approach is needed, one that enables automated, standardized calculation of impact indicators and can be seamlessly integrated into development processes (Dammann et al., 2024; Hunger et al., 2025).



https://doi.org/10.1017/pds.2026.10628 Published online by Cambridge University Press

Given this background, the present work focusses on the ***model-based representation and calculation of LCA-based impact indicators using the Systems Modeling Language (SysML)***. SysML is standardized in the ISO/IEC 19514 (OMG, 2017). The proposed approach targets a consistent integration of quantifiable environmental impact assessment indicators into the early phases of product and system development.

In this approach, *reusable* means that the SysML model elements implementing the approach can be applied within SysML-based models in an MBSE engineering framework independently of the respective product or system, i.e. they can be imported from project to project like a library. *Modular* means that the generalizable parts of the LCA execution, identified from standards and their phases as well as common methods and tool workflows (e.g., indicator definitions, data mappings and calculation rules), are encapsulated as parameterized building blocks, while product-specific inputs are provided separately. *Domain-independent* reflects that parts of the calculation logic are simple, generalizable and partially automatable, enabling modules (e.g. method and indicator selection) to be modeled dynamically so they can be imported across projects and adapted to a specific product or system through configuration rather than remodeling.

Furthermore, by embedding the assessment directly into the SysML-based MBSE workflow and thus into the *model-based development process*, preliminary impact estimates can be generated in parallel with the evolving design models rather than in isolated, retrospective studies. This enables rapid trade-off analyses and iterative improvements and optimizations throughout development, while also supporting post-processing in late-stage design refinement to quantify the resulting impacts before development or production decisions are finalized.

Following this introduction, Section 2 reviews LCA fundamentals and related work in LCA-MBSE integration, including identified research gaps. Section 3 presents the conceptual approach using SysML, Section 4 its technical implementation as a proof-of-concept, and Section 5 summarizes this work, discusses the results and suggests future directions.

## 2. Foundations and related work

### 2.1. Life Cycle Assessment (LCA)

LCA is an internationally standardized methodology for evaluating the environmental impacts of products, processes or services across their entire life cycle. The ISO 14040 standard defines four phases: 1. Goal and Scope Definition, 2. Inventory Analysis, 3. Impact Assessment, and 4. Interpretation (Figure 1).

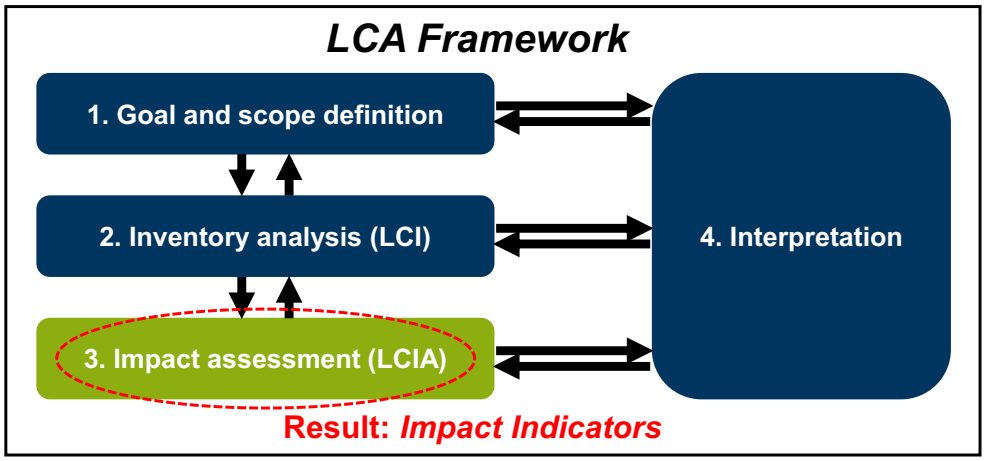



**Figure 1.** Life Cycle Assessment (LCA) framework. Focus on phase 3 — *LCIA*

This structured approach enables evaluations and supports sustainable decision-making. ISO 14044 complements this standard by providing detailed requirements for conducting and documenting LCA studies. Additional relevant standards, such as ISO 14067 (Carbon Footprint) and ISO 14046 (Water Footprint), specify environmental indicators.



https://doi.org/10.1017/pds.2026.10628 Published online by Cambridge University Press

Phases 1 and 2 are highly product-specific. Phase 1 defines the functional unit, goal and system boundaries. Phase 2 compiles the ***Life Cycle Inventory (LCI)***, which represents the raw data basis of an LCA by capturing all relevant material and energy flows within the defined system.
Phase 3, ***Life Cycle Impact Assessment (LCIA)***, translates the LCI data into environmental impacts using characterization factors. These factors quantify the relative contribution of a flow to a specific impact category, e.g. *1 kg* $CH_4$ corresponds to *25 kg* $CO_2$*-equivalents* for global warming potential. The calculation of environmental impacts (LCIA) can be expressed by the following general equation (1), which links inventory data to characterization factors for a specific impact category:

$$Impact = \sum_{i=1}^{n} E_i \cdot CF_i \tag{1}$$

- with $\boldsymbol{E_i}$: the quantity of the i-th emission or resource flow (LCI) and
- $\boldsymbol{CF_i}$: the characterization factor for the respective impact category.

Thus, the calculation requires product-specific inventory data and method-specific characterization factors for different impact categories. Their determination is not defined by the standards but relies on internationally established LCIA methods and databases such as CML, TRACI, Eco-Indicator 99, ReCiPe, ILCD or IMPACT 2002+, differing in scope, geography and scientific models.
A central distinction, e.g. like in ReCiPe (Huijbregts et al., 2017), concerns the use of midpoint versus endpoint indicators. Midpoint indicators ($CO_2$*-eq*, $SO_2$*-eq*) are closer to the emission data, less uncertain, suited for technical optimization in product development. Endpoint indicators aggregate several midpoints into damage-oriented results for human health, ecosystems, and resources, providing intuitive results for decision-making, but with higher uncertainties due to longer and more complex modeling chains.
Finally, Phase 4 interprets the results of the calculated impact indicators, in relation to the original goal and scope to support sustainable decision-making.

## 2.2. Related work on MBSE–LCA integration

MBSE represents system aspects through models, typically combining a modeling language, a tool environment and a method. One of the most widely used language is SysML, enabling structured representation of complex systems. Integrating MBSE with LCA can create synergies by linking system modeling with quantitative environmental impact assessment, offering reusability, automation, consistency, comparability and improved traceability, making environmental impact aspects within engineering processes more accessible and actionable.
Schneider et al. (2023) highlights these potentials in the context of sustainability-related engineering objectives and Advanced Systems Engineering (ASE), emphasizing the added value of holistic, model-based approaches. Several studies have investigated the integration of LCA aspects into MBSE:
Bougain & Gerhard (2017) demonstrate the integration of environmental considerations into SysML models using a 3D printer as a case study. Environmental impacts are linked to components and use-phase scenarios, focusing on the most emission-intensive phases. The approach aims to derive Ecodesign strategies rather than performing a full LCA, and suggests a *future* software implementation (e.g. MagicDraw Plug-in) for automated evaluations. Future extensions propose modeling product variants through "what-if" relationships. Assessments rely on approximate, often subjective scales, limiting transparency and comparability. The authors note that the method is challenging for companies due to its dependence on multiple heterogeneous databases and software tools, tool- and product-specific design, and lack of reusability, underlining existing research gaps.
Following this, Inkermann (2022) systematically reviews approaches to integrating MBSE and LCA, focusing on managing uncertainties and product variants in early design stages. The study particularly analyzes uncertainties in the first two phases of LCA as well as challenges in handling product variants and heterogeneous use cases. As an outcome, an initial concept for methodological and data integration is proposed. While not yet validated, this concept serves as a basis for *future* research directions: Model-based representation of heterogeneous life cycles of components in complex products, development of a modular and product-related LCA, integration of LCA data into SysML models, and the establishment of an LCA-specific modeling profile.
Building on these conceptual contributions, Lipšinić & Pavković (2023) extend this line of research with a case study on a forestry trailer modeled using the MagicGrid approach. SysML diagrams were applied



https://doi.org/10.1017/pds.2026.10628 Published online by Cambridge University Press

to define LCA scope and compare system variants. Compared to earlier work, this study provides a more technical implementation but also reveals major challenges: no integration of LCA databases, high demands on SysML expertise, and overloaded models due to explicit process and parameter modeling. To address these issues, the authors propose for the *future* a SysML profile with new stereotypes separating system and LCA properties. In contrast to Bougain & Gerhard (2017) Ecodesign-oriented approach and Inkermann (2022) conceptual framework, Lipsinić & Pavković (2023) highlight the practical difficulties of achieving robust and reusable MBSE-LCA integration, emphasizing the *need* for standardized modeling guidelines and tool support to improve applicability in engineering practice.

Similar to Lipšinić & Pavković (2023), who rely on the MagicGrid method, Bassam et al. (2024) implement their approach in SysML aligned with the RFLP structure of the V-Model. A humanoid service robot serves as a case study, extending the V-Model to cover the full lifecycle from cradle to grave in early design phases. The study focuses on LCI, while LCIA is not addressed in the SysML model and is regarded mainly as post-processing of LCI results. Since SysML is not designed for LCA calculations, these must currently be performed externally. For *future* work, the authors propose a dedicated SysML profile tailored for LCA and an interface with LCA tools to enable automation. However, the lack of LCIA integration, including impact assessment methods and LCA databases, limits transparency, comparability and reusability, highlighting the *need* for standardized profiles and tool interfaces to broaden MBSE-LCA applicability.

Likewise, Lindemann et al. (2025) pursue the automated generation of LCI data within the SysML context using the Motego method. LCIA is handled externally via an LCA tool, as impact assessment methods are not integrated directly into the system model, highlighting the separation of inventory modeling and impact evaluation.

However, a universally applicable, abstract, and reusable SysML-based approach for LCIA (Phase 3 of LCA), including a consistent representation and computation of impact indicators, is still lacking. Existing solutions are often system- or domain-specific. ***Key challenges*** for SysML-based modeling include dynamic and heterogeneous data availability throughout the development process and the integration of LCA-relevant data with external databases and tools for automated calculations.

Moreover, a range of modularization and parameterization approaches has been proposed within LCA research, yet these concepts are rarely transferred into SysML as reusable libraries or profiles with computable LCIA impact indicator logic. Therefore, this work addresses the **research question**: How can LCIA impact indicators be represented and computed within SysML models to support continuous environmental impact assessment during engineering development? To answer this, we propose a ***reusable, modular, and standards-compliant*** SysML-based approach for LCIA modeling and computation.

## 3. SysML-based concept for LCIA modeling and computation

To address this research gap identified in Section 2.2, namely, the lack of a domain-independent and reusable framework for integrating LCA-based impact indicators into MBSE, this work proposes a **SysML-based modeling approach**. Section 3 introduces the concept for representing and calculating these indicators, focusing on **Phase 3 of the LCA (Impact Assessment)**, while remaining independent of modeling or collecting product-specific inventory data (LCI). The ***goal*** is to embed environmental impact assessment requirements and impact indicators into the system model using a reusable, product-independent approach, linking them to the overall architecture to ensure early, consistent and continuous consideration throughout development, avoid isolated evaluations, and enable comparability, propagation of relevant parameters, and early feedback across all stages.

The modeling method for MBSE applied in this work is not tied to a single technique. Instead, the emphasis is on representing the ***calculation of impact indicators*** within the impact assessment phase (Phase 3 of the LCA) using *SysML*. To embed the concept into the product development process, it is aligned with the RFLP approach (Requirements–Functional– Logical–Physical), which within the V-model enables a continuous and integrated engineering workflow (VDI/VDE 2206). Figure 2 illustrates the developed concept for representing impact indicators by modeling it with SysML.

On the left side of the dotted line, product-specific input data is depicted (LCI, Phase 2), while on the right side, a generic representation of impact indicators (LCIA, Phase 3) and their link to requirements is shown. This representation is independent of concrete products and can therefore be reused across projects. The black frame marks the scope of modeling within SysML models.



https://doi.org/10.1017/pds.2026.10628 Published online by Cambridge University Press

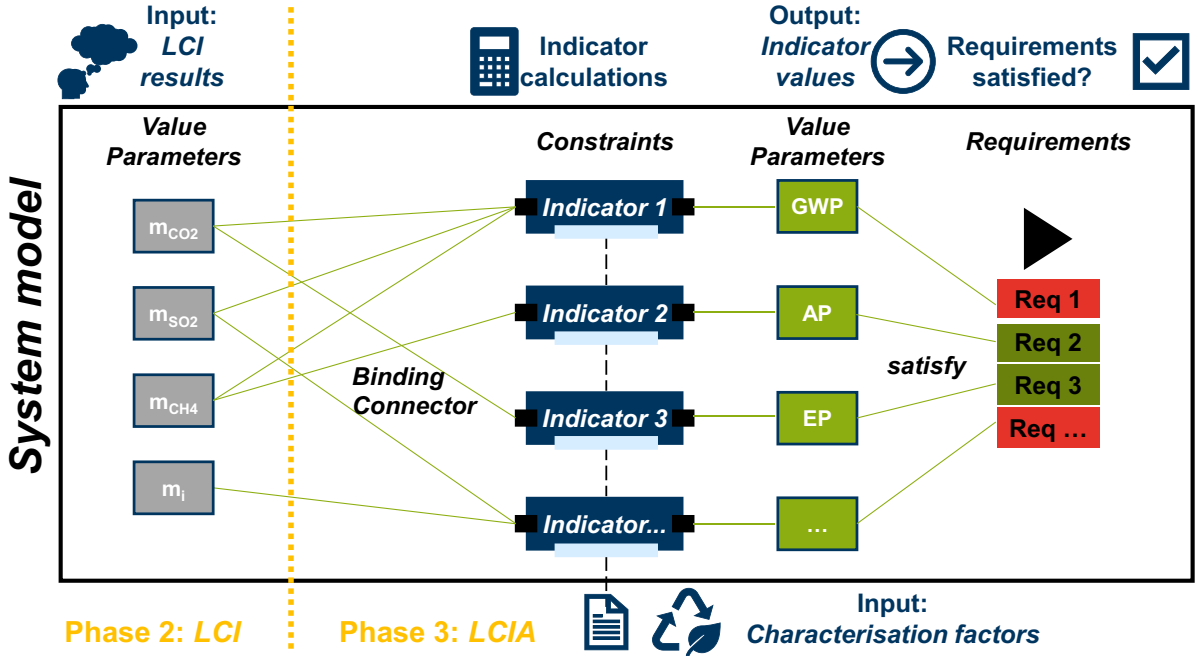



**Figure 2.** SysML-based conceptual schematic for modeling LCA impact indicators

Outside the system model, the calculation of LCIA indicators retrieves the required characterization factors from an external database. These factors are independent of the product and may vary depending on the chosen method, indicator, region or over time, as configuration parameters that drive the retrieval of characterization factors from the external database. Therefore, they are deliberately placed outside the system model, ensuring that the model itself focuses solely on representing the system or product, while the external factors can be updated and maintained independently. This approach focuses on modeling the product-independent parts of the impact indicator calculation, not on the collection of LCI data. Essentially, each value parameter of the LCI results is represented once in the model (single source of truth) and referenced to all calculations in which it is used. This ensures end-to-end data continuity, a core principle of MBSE.

The calculation of impact indicators is modeled with ***parametric diagrams***. For the required *characterization factors*, an external connection via an interface must be defined to retrieve values from corresponding databases or data sheets. Since these factors vary depending on the applied method, impact category, region, or time, the interface must remain flexible and easily maintainable. This setup allows different methods and indicator sets to be integrated without changing the system model itself. The output, the resulting *calculated impact indicators* can then be linked to quantifiable ***requirements*** within the system model, for example by defining limit values. This enables automated verification of requirement compliance.

In the long term, the vision is to represent a configurable set of LCA-based indicators within the system model, allowing engineers to select project or product specific indicators as needed. This is intended to yield a reusable and domain-independent SysML profile and model library that extends beyond individual use cases and supports the systematic integration of environmental impact assessment into engineering processes.

A key advantage of this approach is that LCA-relevant parameters, assumptions, and computed indicators are maintained within the system model and explicitly linked to design elements, and, where necessary, to external data sources. This promotes traceability and consistency and can lower the entry barrier for (systems) engineers, particularly those with limited experience in LCA assessment.

## 4. Technical implementation

To demonstrate the technical feasibility of the proposed conceptual approach, the methodology is implemented in CATIA Magic Systems of Systems Architect using the modeling language *SysML*. SysML provides various diagram types and elements that enable the structured representation of complex systems. In this application, *requirements diagrams, parametric diagrams and block definition diagrams*

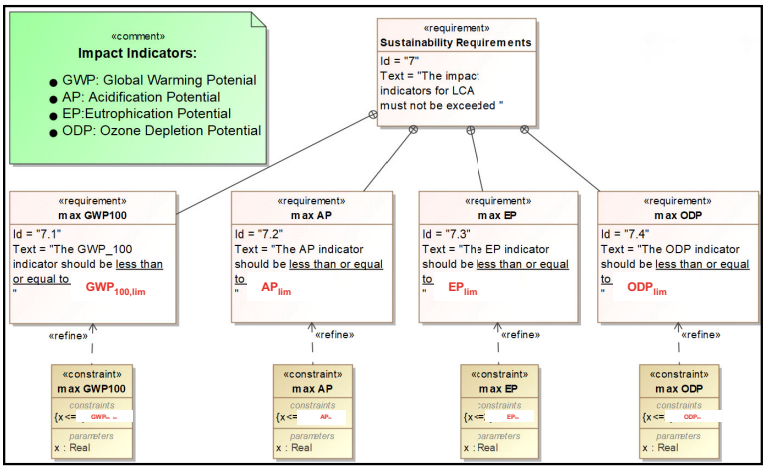


https://doi.org/10.1017/pds.2026.10628 Published online by Cambridge University Press

are combined to model the calculation of impact indicators within ***Phase 3 of the Life Cycle Assessment (LCIA, Impact Assessment)***.
In addition to the elements and diagrams offered by SysML, an *interface* must be developed and implemented, as described in Section 3, to retrieve the data for the characterization factors in a flexible manner. For this purpose, the process was automated using MATLAB, which accesses the characterization factors data stored in an Excel spreadsheet and loads it into the system model.
The proof-of-concept implementation has a demonstrative character. It does not aim to provide a complete product- or data-specific LCA, but rather to illustrate how LCA-based impact indicators can be integrated into a system model in a reusable, domain-independent manner that can be used beyond individual applications. This keeps the approach generally applicable and provides a foundation for future implementations in a concrete development setting with a specific product or system.

## 4.1. Modeling environmental impact requirements

In the first step, requirements are defined and integrated into the system model using ***Requirement Diagrams (req) and Tables***. These requirements represent quantifiable environmental goals (e.g. maximum $CO_2$ emissions per functional unit) and serve as reference values for the subsequent calculation of impact indicators. By linking requirements directly to the system model through a satisfy relationship, traceability is ensured, and compliance can later be automatically verified once the indicators are calculated.
Since the focus lies on environmental factors, requirements are typically formulated as upper limits (e.g. regulatory thresholds), expressed as *"…should be less than or equal to …"*. This formulation ensures quantifiability by defining a constraint value. In SysML, the "≤" relation is automatically interpreted as a constraint and linked to the respective requirement through a refine relationship, enabling the verification of calculated impact indicators by connecting them with the corresponding parameters. Figure 3 illustrates an exemplary implementation in a requirement diagram and its equivalent requirement table.




| # | Id | Name | Text | Refined By | Satisfied By |
|---|---|---|---|---|---|
| 1 | 7 | 7 Sustainability Requirements | The impact indicators for LCA must not be exceeded | | |
| 2 | 7.1 | 7.1 max GWP100 | The GWP_100 indicator should be less than or equal to $GWP_{100,lim}$ | max GWP100 | |
| 3 | 7.2 | 7.2 max AP | The AP indicator should be less than or equal to $AP_{lim}$ | max AP | |
| 4 | 7.3 | 7.3 max EP | The EP indicator should be less than or equal to $EP_{lim}$ | max EP | |
| 5 | 7.4 | 7.4 max ODP | The ODP indicator should be less than or equal to $ODP_{lim}$ | max ODP | |

**Figure 3.** Requirements diagram and table for representing the limitations of impact indicators

A general overarching requirement specifies that the defined LCA impact indicators must not be exceeded. Four subordinate requirements define specific limits for *Global Warming Potential (GWP), Acidification Potential (AP), Eutrophication Potential (EP), Ozone Depletion Potential (ODP)*. The requirement table provides a structured overview of the modeled requirements, in particular highlighting



https://doi.org/10.1017/pds.2026.10628 Published online by Cambridge University Press

the **refine and satisfy** relationships. The latter (*Satisfied By*, last column) will later be connected with the corresponding result parameters of the calculation (See Fig. 4).
In practice, these thresholds can be set by legal regulations or established frameworks. Alternatively, for a manufacturing company, indicator values from previous projects or the current state can serve as references, with lower target values defined to drive sustainable design improvements. The process remains iterative: interim results can be recorded, and trade-offs between requirements considered. Simultaneous consideration of multiple indicators ensures an integrated view, as changes in one single parameter can affect different indicators, enabling a more holistic optimization of sustainable design.

## 4.2. Parametric calculation of impact indicators

The core of the proposed approach is the parametric modeling of LCA-based impact indicators within SysML models. For this purpose, a block called **LCIAlator** (abbreviation for: Calculator of Impact Indicators) is created, which is general and is described by a ***Parametric diagram (par)***. Parametric diagrams capture the mathematical relationships between life cycle inventory (LCI) data (e.g. emissions, resource flows) and characterization factors for the calculation of the impact indicators. Characterization factors are provided dynamically through a MATLAB interface that accesses an external Excel database. This setup ensures flexibility with respect to data source, region, or temporal context while keeping the SysML model consistent and reusable.
Figure 4 depicts the implementation of the proposed approach for embedding environmental impact assessment within an MBSE framework, particularly how LCI data, characterization factors and requirements are seamlessly linked within the system model.
On the left-hand side, the input parameters represent emissions and substances relevant for life cycle assessment, such as sulfur dioxide ($mSO_2$), nitrogen oxides ($mNO_x$), ammonia ($m_NH_3$), and phosphate ($mPO_4$). These parameters are defined once within the system model and reused wherever required. This ensures consistency and eliminates redundant data handling. For example, the parameter $m_{NOx}$ may occur in multiple environmental impact equations, yet it is stored as a single value parameter and connected via binding connectors to the respective constraint blocks.
In the central part of the diagram, the defined LCI parameters feed into dedicated constraint blocks that encode the mathematical formulations of impact indicators. Two exemplary cases are shown: the *Acidification Potential (AP)* and the *Eutrophication Potential (EP)*. Both indicators are computed by linking the LCI parameters with the corresponding characterization factors.
The lower section of the diagram depicts the module for characterization factors. The block is linked to an external process that retrieves characterization values from a structured database of established ***life cycle impact assessment (LCIA)*** methods. This connection ensures that the factors are consistently provided for the subsequent calculations, enabling a transparent and reproducible integration of environmental impact assessment into the system model. Depending on the chosen method (e.g., ReCiPe, ILCD, CML, IMPACT, or TRACI) and the desired indicator (*GWP100, AP, EP, or ODP*), the appropriate factors are automatically retrieved and inserted into the system model.
On the right-hand side, the calculated indicator values are directly linked to requirement elements through *satisfy* relationships. Each requirement specifies quantitative thresholds, for instance, "*The AP indicator should be less than . . .*". Once the indicator values are computed, the system model automatically verifies compliance with the predefined requirements.
This implementation offers several **key advantages**. First, LCI data are defined once and consistently reused across multiple impact assessments, ensuring coherence and avoiding redundancy. Second, the integration of characterization factors is automated and flexible with respect to the chosen assessment method, which minimizes manual effort, reduces the risk of errors, and guarantees that the model reflects the most current methodological choices. Third, indicators are directly linked to system-level requirements, enabling continuous validation within the design process. Together, these capabilities transform assessment from a separate, one-time analysis into an integral component of the iterative MBSE workflow. As a result, the approach provides a robust framework for embedding environmental impact performance objectives systematically into model-based product development.



https://doi.org/10.1017/pds.2026.10628 Published online by Cambridge University Press

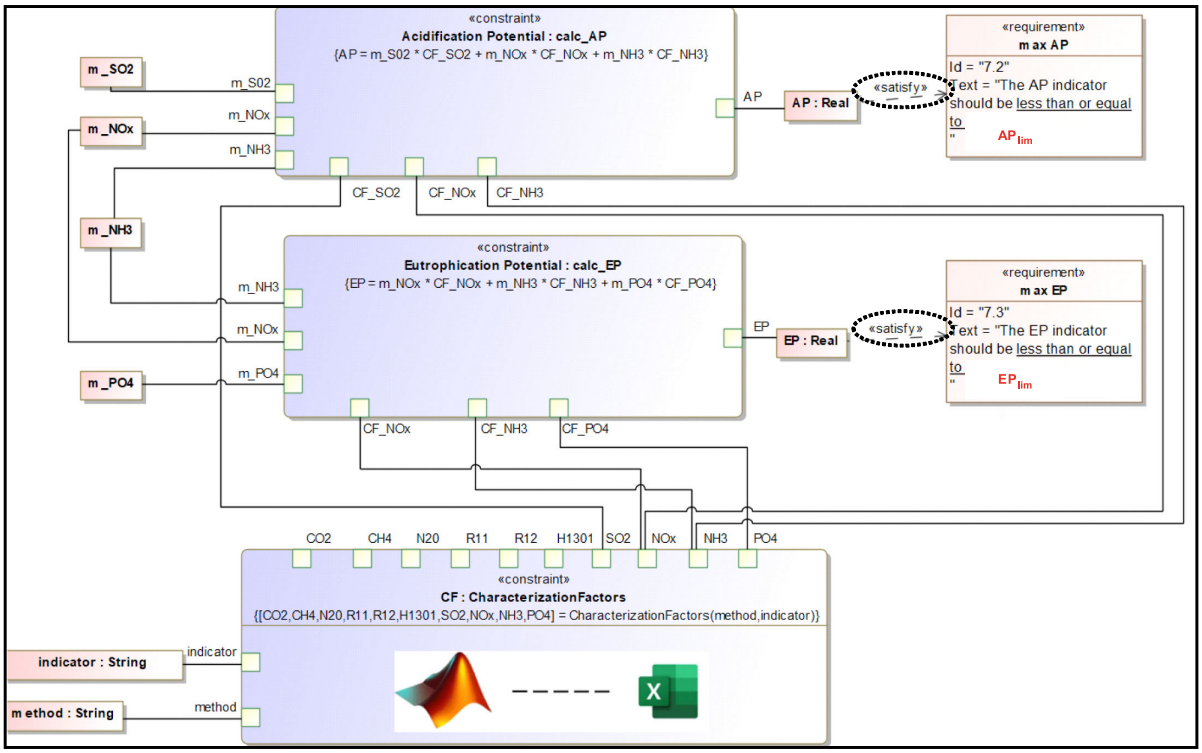



**Figure 4.** Parametric diagram for calculation of impact indicators and link to requirements

## 4.3. Integration into the product structure and its components

Building on the parametric setup introduced in Section 4.2, the calculated impact indicators are now integrated into the product architecture. This is achieved in a ***block definition diagram (bdd)***, where the block called *LCIAlator*, containing the calculations and value parameters described previously, is connected to the whole product architecture (Figure 5).

Through a generalization relationship, the *LCIAlator* is associated with the components so that all necessary LCIA attributes are inherited. The product architecture itself is represented using composition relationships, which describe the hierarchical decomposition of the product down to individual components. In practice, all blocks requiring LCIA assessment are generalized with *LCIAlator*, ensuring automatic access to the parametric setup from Section 4.2. If now a diagram is created for a block *component*, the LCIA calculations are directly inherited and can be calculated.

To this end, the combination of a *single, abstract LCIAlator block* with the hierarchical structure of assemblies and components establishes a reusable LCIA integration mechanism that works independently of a specific product. Since the LCIA logic is modeled once at a generic level and inherited through generalization, any product system can obtain full LCIA capability without rebuilding the computational model. This *significantly increases modeling efficiency, reduces implementation effort and avoids inconsistent or duplicated LCIA logic across projects*.

Furthermore, the *LCIAlator* block can be profiled, library-based or distributed as a modeling asset across different *SysML projects*. This enables seamless transfer of LCIA functionality across domains, product generations, and development teams, while preserving the consistency of indicator definitions, calculation methods and characterization factors.

As a result, environmental impact aspects based on LCA can be addressed *systematically, repeatedly and at early design stages*, rather than treated as isolated one-time analyses detached from engineering. The LCIA capability becomes an inherent element of every system model that uses this framework integration, ensuring that *environmental impact indicators remain computable, comparable, and traceable throughout the whole development lifecycle*.



https://doi.org/10.1017/pds.2026.10628 Published online by Cambridge University Press

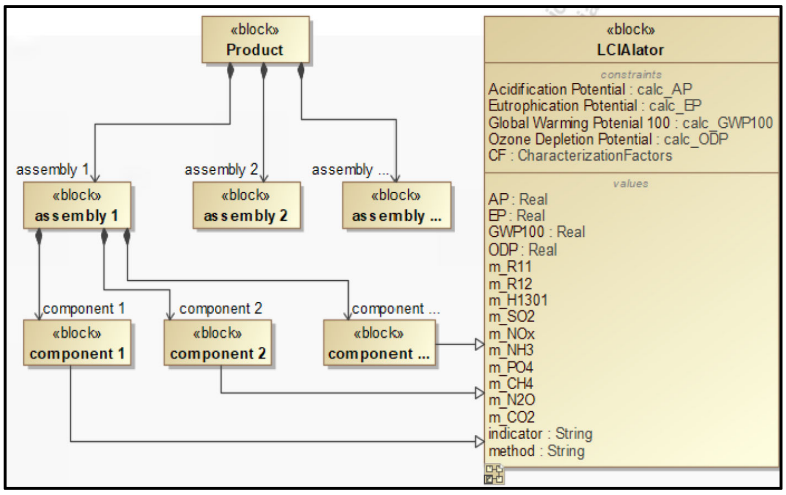



**Figure 5.** Block definition diagram for product structure and inheritance from the block LCIAlator

## 5. Summary & outlook

This work addresses the integration of environmental impact aspects into early product development by embedding **Life Cycle Assessment (LCA) with Model-Based Systems Engineering (MBSE)**. The analysis in Section 2 demonstrates that LCA studies, although standardized and internationally recognized, are often time-consuming, fragmented, and only partially reusable in practice. In particular, conventional LCA methods are often complex and difficult to integrate consistently into early-stage product development workflows. ***Phase 3 of LCA (Impact Assessment, LCIA)*** is rarely embedded in a systematic and reusable manner, and current approaches are typically product-specific, conceptual, or focus solely on *inventory data (LCI)*, limiting comparability and traceability across systems and projects. This paper introduces a ***model-based approach using the Systems Modeling Language (SysML) to integrate LCA-based impact indicators into product and system design early in the development process***. By embedding assessments within MBSE, the method ensures *consistent, traceable, and automated evaluation of environmental impacts*. The proposed approach provides *reusable SysML-based modeling structures* that link abstract representations of *LCIA* directly to system models, enabling product-wide consistency, avoiding isolated evaluations, and maintaining independence from modeling or collecting product-specific inventory data. Requirements and impact indicators are modeled parametrically, creating a coherent framework that supports decision-making, iterative validation, and the identification of design opportunities to improve environmental impact performance throughout the product lifecycle.

The *technical implementation* shows how impact assessment indicators can be linked to system-level requirements and automatically verified within a model-based workflow. The framework is ***domain-independent*** and designed to be *scalable* across different products, systems, and engineering teams, promoting *transparency, reusability, and acceptance* among engineers, including those with limited LCA expertise. A *key limitation* of the current implementation is that the general LCIA calculation equation is essentially static and does not dynamically adapt to variations, due in part to the limitations of SysML v1, which are expected to be addressed by the *standardized API* in ***SysML v2***.

Overall, this work provides a **systematic, model-based methodology for integrating LCA into early product development**, demonstrating how ***MBSE*** can support ***environmental impact assessment*** objectives. The paper presents a reusable, modular, and domain-independent MBSE framework for representing and computing LCA impact indicators and demonstrates its technical feasibility through a proof-of-concept implementation within SysML models. Limitations and potential extensions outline opportunities for future research toward more automated and standardized LCA-based impact assessments in early design phases. A comprehensive validation is planned as future work once the implementation is applied in a concrete development setting.

## Acknowledgement

This research was partially funded by the project #dzr | digital.zirkulär.ruhr, which is supported by the ERDF/JTF NRW 2021-2027 program.



https://doi.org/10.1017/pds.2026.10628 Published online by Cambridge University Press

## References


Bassam, H., Lünnemann, P., Riedelsheimer, T., & Lindow, K. (2024). A Model-Based Methodology for Life Cycle Assessment from Cradle-to-Grave Early in Product Design. *Procedia CIRP*, 128, 662–667. https://doi.org/10.1016/j.procir.2024.07.058

Bougain, S., & Gerhard, D. (2017). Integrating Environmental Impacts with SysML in MBSE Methods. *Procedia CIRP*, 61, 715–720. https://doi.org/10.1016/j.procir.2016.11.196

Ciroth, A. (2007). Ict for environment in life cycle applications openLCA — A new open source software for life cycle assessment. *The International Journal of Life Cycle Assessment*, 12(4), 209–210. https://doi.org/10.1065/lca2007.06.337

Dammann, M. P., Hoffeins, P., Barth, J., Saske, B., Paetzold-Byhain, K., & Modler, N. (2024). Modeling a sustainability-oriented development process with model-based systems engineering, 75–84. https://doi.org/10.35199/dfx2024.08

Diaz, A., Schöggl, J.-P., Reyes, T., & Baumgartner, R. J. (2021). Sustainable product development in a circular economy: Implications for products, actors, decision-making support and lifecycle information management. *Sustainable Production and Consumption*, 26, 1031–1045. https://doi.org/10.1016/j.spc.2020.12.044

Finkbeiner, M., Ackermann, R., Bach, V., Berger, M., Brankatschk, G., Chang, Y.-J., Grinberg, M., Lehmann, A., Martínez-Blanco, J., Minkov, N., Neugebauer, S., Scheumann, R., Schneider, L., & Wolf, K. (2014). Challenges in Life Cycle Assessment: An Overview of Current Gaps and Research Needs. In W. Klöpffer (Ed.), *LCA Compendium – The Complete World of Life Cycle Assessment. Background and Future Prospects in Life Cycle Assessment* (pp. 207–258). Springer Netherlands. https://doi.org/10.1007/978-94-017-8697-3_7

Huijbregts, M. A. J., Steinmann, Z. J. N., Elshout, P. M. F., Stam, G., Verones, F., Vieira, M., Zijp, M., Hollander, A., & van Zelm, R. (2017). ReCiPe2016: a harmonised life cycle impact assessment method at midpoint and endpoint level. *The International Journal of Life Cycle Assessment*, 22(2), 138–147. https://doi.org/10.1007/s11367-016-1246-y

Hunger, T., Arnold, M. G., Engesser, S., & van den Gerald Boogaart, K. (2025). Integrating sustainability facets into the early stages of new product development using multi-criteria tools - a critical review. *Journal of Cleaner Production*, 517, 145834. https://doi.org/10.1016/j.jclepro.2025.145834

INCOSE (2021). Systems Engineering Vision 2035, Article.

Inkermann, D. (2022). Potentials of integrating MBSE and LCA to handle uncertainties and variants in early design stages, 10. https://doi.org/10.35199/dfx2022.19

ISO 14040. *DIN EN ISO 14040, Environmental management - life cycle assessment - principles and framework (ISO 14040:2006 + Amd 1:2020)*. Beuth Verlag GmbH.

ISO 14044. *DIN EN ISO 14044, Environmental management - life cycle assessment - requirements and guidelines (ISO 14044:2006 + Amd 1:2017 + Amd 2:2020)*. Beuth Verlag GmbH.

ISO 14046. *DIN ISO/TR 14073, DIN SPEC 35806, Environmental management - water footprint - illustrative examples on how to apply ISO 14046 (ISO/TR 14073:2017)*. Beuth Verlag GmbH.

ISO 14067. *DIN EN ISO 14067, Greenhouse gases - carbon footprint of products - requirements and guidelines for quantification (ISO 14067:2018)*. Beuth Verlag GmbH.

Lindemann, M., Jacobs, G., Dreier, J., Wischmann, S., Höpfner, G., & Berges, J. (2025). Facilitating LCA During Product Development by Automated Inventory Model Generation Using SysML System Models. *Sustainability*, 17(11), 4783. https://doi.org/10.3390/su17114783

Lipšinić, Z., & Pavković, N. (2023). Integrating life cycle assessment in model-based systems engineering. Advance online publication. https://doi.org/10.22032/DBT.58905

OMG (2017). ISO/IEC 19514: Information technology — Object Information technology — Object management group systems modeling language (OMG SysML).

Popowicz, M., Katzer, N. J., Kettele, M., Schöggl, J.-P., & Baumgartner, R. J. (2025). Digital technologies for life cycle assessment: a review and integrated combination framework. *The International Journal of Life Cycle Assessment*, 30(3), 405–428. https://doi.org/10.1007/s11367-024-02409-4

Rotondo, B., Bakker, C., Balkenende, R., & Arquilla, V. (2025). Integrating Circular Economy Principles in the New Product Development Process: A Systematic Literature Review and Classification of Available Circular Design Tools. *Sustainability*, 17(9), 4155. https://doi.org/10.3390/su17094155

Schneider, B., Spindler, H., Kürümlüoglu, M., & Riedel, O. (2023). Nachhaltigkeit durch Advanced Systems Engineering/Advanced Systems Engineering approaches for the creation of sustainable systems - Sustainability through Advanced Systems Engineering: Lösungsansätze des Advanced Systems Engineering für die Gestaltung nachhaltiger Systeme. *Wt Werkstattstechnik Online*, 113(06), 214–218. https://doi.org/10.37544/1436-4980-2023-06-6

Teixeira, N. (2025). Circular economy perspectives: challenges, innovations, and sustainable futures. *Discover Sustainability*, 6(1). https://doi.org/10.1007/s43621-025-01606-x

VDI/VDE 2206. *VDI/VDE 2206, Development of cyber-physical mechatronic systems (CPMS)*. Beuth Verlag GmbH.




https://doi.org/10.1017/pds.2026.10628 Published online by Cambridge University Press