Proceedings of the Design Society, Volume 6: DESIGN 2026
https://doi.org/10.1017/pds.2026.10644

Hierarchical lifecycle modelling in circular product development

Marie Schwahn[1,✉] **and Lukas Block**[2]

[1] *Institute of Human Factors and Technology Management (IAT), University of Stuttgart, Germany,*
[2] *Fraunhofer Institute for Industrial Engineering IAO, Germany*

✉ marie.schwahn@iat.uni-stuttgart.de

ABSTRACT: As companies face increasing accountability for resource efficiency, circularity measures like repair or remanufacture offer a promising solution. Modeling expected product lifecycles in early design phases is crucial for planning their effective application. This paper introduces a hierarchical lifecycle model to represent component-specific lifecycle paths across all product architecture levels in a single model. Our approach ensures consistency across hierarchy levels and facilitates precise application of component-specific circularity measures, promoting effective circular product design.

KEYWORDS: circular economy, modelbased systems engineering (MBSE), structural complexity, early design phase

1. Introduction

Companies are increasingly held accountable for the resource-efficiency and ecological sustainability of their products (Adamowicz, 2022). The concept of a circular economy offers one solution to this challenge. It describes the idea of moving to a circular form of value creation, for example by repurposing, recycling or remanufacturing components or materials of products, instead of disposing them at the end-of-life (Ellen MacArthur Foundation, 2013). Yet, products need to be designed in a way, so that they can be easily circulated by applying so-called R-Strategies as defined by (José Potting et al., 2017) - also referred to as circularity measures within this work. The early phases of product design (like fuzzy frontend and concept phase) are fundamental to developing such circular products. Within these phases, designers and engineers still have high degrees of freedom that - on the one hand - allow for the full exploitation of opportunities to optimize a product's design towards circularity. On the other hand, making suboptimal decisions during these phases often leads to missed potential in the sustainability and circularity of the final product over its lifetime. For example, a repair may no longer be feasible at later stages if a defective component is not designed to be easily accessible or removable. Similarly, recycling cannot be applied if the respective component is manufactured from a non-recyclable material. Design decisions made during product development therefore directly determine the applicability of RStrategies throughout the product lifecycle.

Yet, in early phases of product development, there also exists a high degree of uncertainty regarding the sustainability implications of design decisions (eco-design paradox (Bhander et al., 2003)). This uncertainty is grounded in a yet-not-complete product design and an (at least partially) unknown lifecycle of the product. Furthermore, complex trade-offs and rebound effects between the product architecture and the lifecycle exist, making design decisions in those early phases non-trivial. Therefore, it is essential to not only consider the static product architecture, but also the expected lifecycle of a product in early design stages including all possible lifecycle paths of a product and its components with their likelihood. Based on the expected lifecycle, it becomes possible to evaluate whether the application of circularity measures in product design - like remanufacturing or repair, … - is really beneficial and improves circularity. Thus, different authors developed approaches to model expected lifecycles in order to assist

https://doi.org/10.1017/pds.2026.10644 Published online by Cambridge University Press

product development. Such approaches have demonstrated to promote systemic thinking and efficient circular product development for given challenges (see e.g. (Schwahn et al., 2024), Section 2).
However, existing lifecycle modelling approaches often do not consider the hierarchical structure of product architectures: Typically, a product is comprised of different layers of modules and child components. The lifecycles of these modules and child components are different but interrelated to each other and to the product's overall lifecycle. Those individual lifecycles of modules of the product must be consistent with each other at any time during the lifecycle - for example a child component cannot be recycled while its parent component in which it is integrated is still modelled to be in use. Within this work, our focus lies first and foremost on ensuring consistency between the individual lifecycle trajectories of components throughout the lifecycle of a given product instance, rather than on establishing compatibility between different product or component generations in the case of a reintegration. However, previous probabilistic lifecycle models applicable for early design phases of circular product development where uncertainties are present are designed to represent the expected states and lifecycle trajectories at a single hierarchical level - the level of the overall product. This results in several challenges: (1) circularity measures - like remanufacturing, repurposing, repairing or partial recycling - cannot be modelled correctly, because components of the product leave the primary product. Thus, the required processes (i.e, steps in a lifecycle) to implement such strategies do require the specific treatment of child components and the tracking of independent pathways for further analysis, which is not possible with present lifecycle models. For example, when a child component leaves the system of the product under development through the application of a reuse, the following individual path of this component is relevant to understand whether the component returns to the system and can be reintegrated into the incomplete parent module. (2) Lifecycles of interrelated components within one product structure must be modelled independently with current approaches. This can lead to inconsistencies between lifecycles of different hierarchy levels of a product - leading to inaccurate evaluations of the product's circularity (e.g. in lifecycle assessments (LCA)). The independent modelling of lifecycles could for example result in two lifecycles of a parent and child component, where the child component is still modelled to be in the production phase, while the parent component is already assumed to be in use - this is a practically impossible scenario where parent and child component paths are not well aligned. Furthermore, this can result in wrong assumption about the lifecycle requirements towards a component. (3) The application of circularity measures to a product can only be modelled and planned for the product as a whole, not for a specific child component. This can lead to designs where circularity measures are not applied to their most specific - and thus beneficial - component. For example, a whole module is replaced or refurbished while only child components of the product are defective or malfunctioning. In this case, product components are devalued or are subject of labour- and resource-intensive processes even if it would not be necessary.
The objective of this work is therefore to present a new model counteracting the limitations in previous modelling approaches to provide a robust and well-founded basis for an MBSE-based approach to circular product development. The aim was accordingly to develop a model that effectively addresses three persistent challenges: correct representation of circularity measures and their implications for child components, consistent modelling across hierarchy levels, and targeted, component-specific application of circularity measures. We propose a novel, hierarchical approach for lifecycle modelling with circularity measures that overcomes those challenges. It allows to model individual lifecycles for components on different hierarchy levels of a product while ensuring consistency and allowing for the effective and precise integration of component-specific circularity measures. Accordingly, this work primarily addresses stakeholders involved in product development by supporting a deeper understanding of the challenges associated with hierarchical lifecycle modelling and the benefits of the proposed approach. The model provides a robust foundation for developing decision-support tools intended for designers, engineers, product developers, and other practitioners engaged in early-stage product development. In addition, this work provides guidance for researchers working with MBSE-based or related modelling approaches in circular product development. It outlines the key modelling elements required to build an MBSE-based decision-support framework for early design phases.

2. Literature review

Life Cycle Engineering (LCE) comprises engineering strategies aimed at designing and manufacturing products in ways that optimize their overall lifecycle performance and sustainability (Jeswiet, 2014). Consequently, improving a product's circularity represents one aspect of LCE. Many sources identify

https://doi.org/10.1017/pds.2026.10644 Published online by Cambridge University Press

Model-Based Systems Engineering (MBSE) as a promising approach for addressing the challenges inherent in LCE and especially circular product development (Bougain & Gerhard, 2017; Dér et al., 2022; Halstenberg et al., 2019; Yvars & Zimmer, 2021). MBSE approaches are traditionally composed of a modelling language, a method and a modelling tool – for example a software tool (Sanford Friedenthal, 2008). The underlying modelling language helps to create a formalized representation of the system under development with a clear structure.

Thus, different authors, have addressed the modelling of expected product lifecycles in early phases of product design. Block et al. (2023) for example present an approach to model probabilistic lifecycles feasible for early design phases - see Figure 1. The lifecycle models are designed as a form of stochastic petri nets. Lifecycles are represented as a progression of states that a product goes through from beginning to the end of life. Probabilistic transitions between the states indicate the likelihood of one state following on the previous state. This modelling approach captures lifecycle uncertainties present in early design phases, where the lifecycle of different product instances may still be uncertain or statistically distributed. The modelled uncertainties can then be documented. Their impact can be considered when predicting impacts of design decisions on the product's lifecycle.

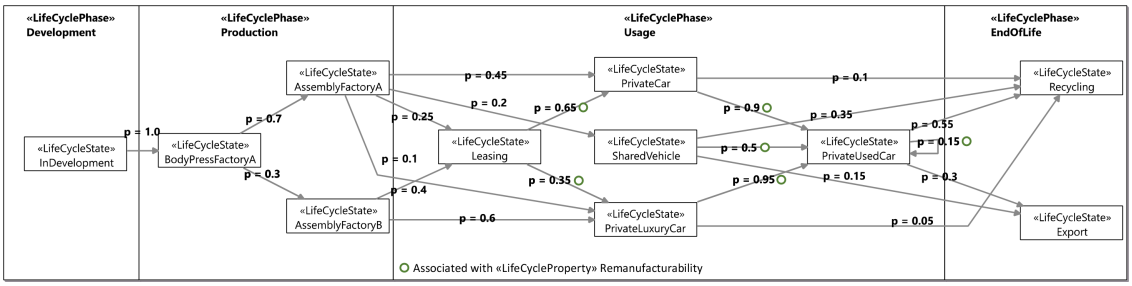


Figure 1. Lifecycle model as a progression of states connected by probabilistic transitions (Block et al., 2023)

Schwahn et al. (2024) extend this approach of modelling probabilistic product lifecycles by the concept of integrating tangible circularity measures. Circularity measures are tangible and product related measures based on the R-Strategies defined by Potting et al. (2017) which can be directly included into a modelled product lifecycle in a specific state. Yet, both approaches, (Block et al., 2023) and (Schwahn et al., 2024) only model probabilistic lifecycles for one product, ignoring the hierarchical structure of a product with modules, components and parts.

Beyond these models, the literature offers several further approaches to lifecycle modelling. (Shu & Wang, 2007) introduce an approach in which lifecycle models are defined as a combination of multiple individual models, partly storing information about the static product and the Bill of Material (BoM) and partly also providing information about sequential processes. When combined, these models provide the required information for all lifecycle phases. (Anand & Wani, 2010) define an approach for product lifecycle modelling and design by using a digraph to model a set of Life Cycle Design (LCD) attributes such as performance, maintainability, and safety, and their relations to each other. However, while both approaches address lifecycle modelling and design, they do not provide an option to model the complete product and component dynamics from production to end of life in a sequential lifecycle model. (Wouterszoon Jansen et al., 2020) introduce such an approach to model product and component dynamics over the complete lifecycle for cost-optimization purposes. However, the presented approach stipulates that the lifecycle of each child component of the product is modelled in a separate model instance, and only the analysis results - in this case the cost analysis - of all individually modelled lifecycles are combined. Consequently, modelling within a single integrated model - one that depicts dependencies and ensures consistency across the product's hierarchical levels - is not possible with this approach. Such an integrated lifecycle modelling approach for a product and all its child components is presented by Umeda et al. (2012). However, the presented approach for lifecycle modelling is not based on probabilistic modelling which is essential in the early phases of product development to consistently represent uncertainty and probability-based possible paths for a product and all its components. In addition, states cannot be nested, preventing the clustering and hierarchical representation of child states and more detailed underlying processes.

In conclusion, existing approaches provide a valuable foundation for lifecycle modelling, but they exhibit shortcomings - either by separating product and component lifecycle models, resulting in possible inconsistencies between hierarchy levels, or by lacking probabilistic which is essential in early design

https://doi.org/10.1017/pds.2026.10644 Published online by Cambridge University Press

phases, as well as hierarchical capabilities to structure processes. An approach for modelling of individual lifecycle paths for child components of the product in one integrated lifecycle model, ensuring consistency across hierarchy levels while accounting for uncertainties in early design phases and relationships between states and their child states, has not yet been introduced. Consequently, to address these limitations, we propose a hierarchical probabilistic lifecycle modelling approach that integrates product and component lifecycles into one consistent model. The approach builds upon existing modelling approaches but additionally reflects the hierarchical structure of product architectures and captures the individual lifecycle behavior of each component. It also applies the concept of tangible circularity measures at the component level. This allows for consistent, component-specific modelling and evaluation of circularity strategies under uncertainty in early design phases.

3. Approach

Our approach extends Block et al.'s (2023) idea of probabilistic lifecycle modelling: We understand a lifecycle as a probabilistic sequence of states, depicting possible paths that the different product instances might follow. We extend this approach by reflecting the hierarchical structure of a product: For each layer in the product architecture's hierarchy, we introduce a related hierarchy level in the lifecycle model. Furthermore, the concept of integrating circularity aspects into lifecycle models via tangible circularity measures as described in Schwahn et al. (2024) is added to this model.
The model is developed starting from a stochastic petri net representation of expected product lifecycles. Lifecycle states are systematically linked to elements of the product architecture through contextual assignments and organized using nested state structures. This ensures consistency across hierarchy levels while enabling the explicit representation of component-specific processes. Circularity measures are then integrated as dedicated lifecycle states within this hierarchical structure, resulting in a unified probabilistic lifecycle model.

3.1. The concept of hierarchical probabilistic lifecycles

(Block et al., 2023) describe a lifecycle as a probabilistic sequence of states, depicting possible paths that the different product instances might follow (see Figure 1). This approach is extended in two ways to achieve lifecycle models, considering different but consistent product hierarchies.
First, we introduce the new idea of a ***context*** for each state in a lifecycle. A context defines to which components of a product architecture a certain state belongs, i.e. which modules, components or parts traverse this state and are affected by it. For example, a modelled element such as the center console module might be disassembled to perform an R-Strategy to access and repair a single broken child component. This is depicted in Figure 2. The context of the state before the disassembly is the complete *CenterConsoleModule* (see yellow marking), while the context of the following repair state where only relevant to the side covers is *Sidecovers* (see purple marking). The context of the state representing the cleaning of the remaining console has the context of all remaining child components of the *CenterConsoleModule* (see blue marking). The context is therefore modelled as an explicit attribute of a state referring to one or multiple elements of the product architecture. It should be noted that, the reference to a component always applies to the component itself as well as all of its child components (see yellow marking to *CenterConsoleModule* automatically including all its child components). An explicit enumeration of components is only necessary when referring to components that do not share a hierarchical parent–child relationship. Consequently, a context links lifecycle states to specific components at any hierarchical level of the modelled product architecture, allowing the representation of different component-specific states within a single model (Figure 2). This allows realistic modelling where child components follow independent paths, common in manufacturing and circular economy scenarios. For example, in the circular economy related scenario a malfunctioning component may be disassembled and undergo repair or remanufacture individually without affecting the entire product. This enables targeted and resource-efficient modelling of circularity measures. The application of a circularity measure, which may be energy- or material-intensive, does not need to be modelled and implemented for the complete product but solely to the defective component in question. To keep the modelling effort within a practicable scope, the assignment of contexts can be performed semiautomatic. Linearly successive states always inherit the same context as their predecessor states, and substates of a parent state likewise share the context of that parent state; in these cases, the context can be assigned automatically. Only when a disassembly occurs do individual paths of different disassembled child

https://doi.org/10.1017/pds.2026.10644 Published online by Cambridge University Press

components diverge. To reflect this, the context of the following states on the individual paths must change - the new context must be specified manually.

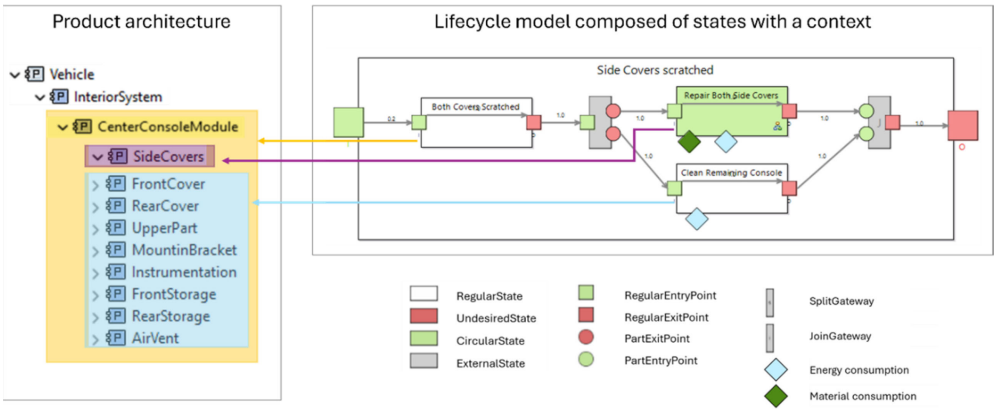


Figure 2. Relations between different levels of the hierarchical product architecture and individual states within the lifecycle model as defined by state contexts

Second, we introduce the concept of ***nested states*** for our lifecycle model: Lifecycle states may now contain other states, so-called child states. In product lifecycles, processes rarely occur strictly in sequence; many states consist of multiple underlying subprocesses. These subprocesses may describe more detailed steps for the same product or the behavior of a specific child component within a broader state. For example, a production phase typically includes several individual manufacturing steps, which we model as child states. Child states therefore either refine the parent state or represent component-specific subprocesses. This nested structure clarifies the relationship between higher-level and subordinate processes and can also represent broader lifecycle phases—such as the use phase—in which concrete processes take place. Depending on their function, child states may refer to the same architectural level or to lower-level components. As a result, the modelling approach provides both detailed process descriptions and explicit subprocess definitions for child components. Thereby, the nested states establish consistency between lifecycles across hierarchy levels by design. It is not possible for a child component to remain in a certain phase or higher-level state of the lifecycle such as the production while its parent component is already in another phase such as the usage. All production related states of the child component are contained in the production state of the parent component. A child component can, however, deliberately leave a parent state via a special exit point of this state. This automatically indicates that the component is following its own independent path, separate from the parent component for the moment. Additionally, the concept of nested states allows to structure the complex state sequences by clustering interrelated states. It provides a good overview of the system dynamics on different levels of detail.

As described in Section 2, our basic modelling approach is designed as a type of stochastic petri net with places, transitions with likelihoods and tokens. In the hierarchical lifecycle model, the tokens traveling through the net represent instances of the product and of its different child components on all hierarchy levels. The paths of the individual components are modelled via separate tokens that might be joined into a module or the overall product or might be split again during the lifecycle. Thus, a realistic representation of lifecycles requires the model to allow multiple possible options of behaviour. Singular, straight-forward state sequences for the product as a whole, or parallel processing for different child components of the product. To do so, we define two types of places where firing and transitioning of tokens happens based on a different logic and requirements.

The first type or places are states. States only allow consistent incoming and outgoing transition: The group of instances traversing a state can only enter together from one of the possible incoming paths and take one of the outgoing transitions all together. They do not split up or join paths. For this type, the context of the state defines which group of components can enter and leave it. This means a state which is by context relevant for the product as a whole can only be entered by an instance (token) representing the entire product, while a state with a context of only a child component can only be entered by this specific child component (i.e. the representing token). This requirement is realized

https://doi.org/10.1017/pds.2026.10644 Published online by Cambridge University Press

through so-called regular entry points and regular exit points of a state (see e.g., Figure 4). Yet, there can be multiple, but alternative incoming and outgoing transitions.
The second type of places are new: So-called Gateways. Gateways represent splits or joins and thus either the incoming or the outgoing paths are parallel paths where multiple tokens can leave or exit at the same time. We differentiate between two types of gateways. Join gateways have multiple incoming paths. Different tokens (component instances or groups of components) can be synchronized here to merge individual paths before traversing a shared state referencing them all together. Thus, join gateways combine instances of multiple child components and their individual paths into a single path. They consume multiple tokens and generate a common token representing the sum of joined components. Split gateways divide a component or group of components into separate paths. At a split gateway, a single input token is consumed, and separate tokens are generated for each branching child component, enabling independent paths to emerge from a shared one. A gateway's context represents the sum of the combined/divided components. Part entry points or part exit points specify which individual components (tokens) are required to enter the join gateway to be merged or exit the split gateway to follow individual paths. These points have a context on their own, which must present a subset of the context of the containing state. Part entry points or part exit points are additionally used to model the case where a subset of the context of a state, leaves or enters the state. This can be the case, when a produced child component bypasses the assembly in the production for example, and is instead transferred to the use state where it is used to replace a malfunctioning component. In this case not the product as a whole exits the production state, but only the respective child component.
Consequently, the lifecycle model is designed, so that it ensures the correct simulation of lifecycles. For each product instance represented as a token within the Petri net, an individual deterministic lifecycle path with all included child component paths emerges when firing the petri net. If a component instance is not available at a certain point but required for the further path of the currently incomplete product the simulation cannot continue. The firing condition at the corresponding place - e.g., a join gateway - is not fulfilled. This can be the case either because it has not yet been initially produced or because the component has exited the system through a circularity measure (e.g., reuse, see Figure 3). To prevent this, tokens for necessary child components are automatically spawned at possible incoming individual lifecycle paths to generate the required child component instances for a specific path. In Figure 3 a new instance is added from an external path generating the new child component instance. This allows for an automated model-based simulation of individual deterministic product lifecycles, including their child components. In one simulation run, the lifecycle of a single product instance (a token) is simulated. Based on the rules for firing the network, which are realized through entry and exit points, it is automatically ensured that required child components are sampled correctly. Thus, the consistency between required sub-paths and the path of the product instance itself, as well as the completeness of the product, is automatically ensured. By running multiple, probabilistic simulations, different lifecycle paths can be sampled, based on the transition probabilities.
Yet, this consistent approach to product lifecycle paths does not only allow to model detailed child processes in the production but specifically enhances the potential of modelling and planning targeted circular economy strategies. The hierarchical modelling approach allows to integrate circularity measures as introduced by (Schwahn et al., 2024) for specific critical components in the product architecture; for example the repair of a broken child component, the reuse of a module composed of multiple child components or the recycling of the set of all child components which are made of recyclable materials at the end of life. Circularity measures are characterized by the fact that they often address the removal, refurbishment, and reintegration of child components.
Figure 3 depicts the exemplary case of a product instance leaving the system via a circularity measure such as a reuse. A component leaves the system - as described in the previous section. The product can continue only once a replacement component is provided. In such a case an individual lifecycle path can be modelled, consistency can be checked and it can be simulated, where a new component instance originating from the production or another system joins the product. Therefore, circular economy can be applied more focused and effective through our modelling approach without wasting energy or material on higher levels components which are not the source of a malfunction of a parent module.

https://doi.org/10.1017/pds.2026.10644 Published online by Cambridge University Press

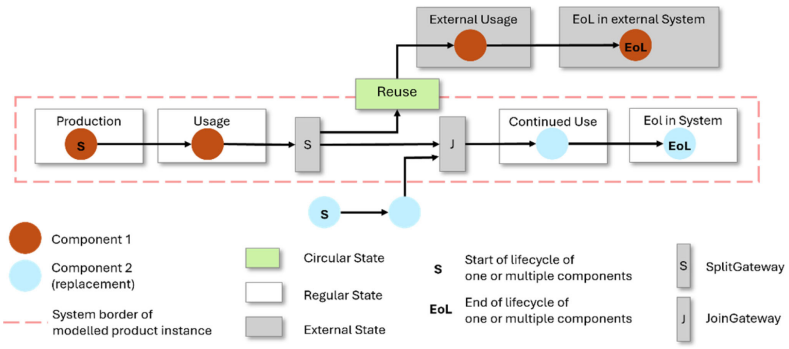


Figure 3. Exemplary deterministic path of a product instance based on the hierarchical lifecycle model with two instances of the same component visiting the system (Fraunhofer, 2025)

3.2. Model application

The hierarchical model itself (see Section 3.2) is agnostic towards the chosen design process (e.g., waterfall, iterative, . . .). Yet, in evaluation practice (see Section 4), we saw that it works well with an iterative approach that considers the increase in knowledge during the design process, because the application of the model is intended to already take place during early development stages. At first expected lifecycles for the product are modelled to analyse how the product will behave throughout its lifetime. Based on this, further design decisions with regards to the circularity of child components can be taken effectively. The product is being further defined stepwise at the level of modules, components, and parts. Consequently, the lifecycle model is extended simultaneously because the modelling of the lifecycle layers requires that the associated level within the product architecture exists.

Due to the different degrees of knowledge during product development, modelling can be approached in two ways: Firstly, it can be fully based on assumptions and estimates in case the modeller does not have any prior knowledge of how the product will behave throughout the lifecycle. Estimates and assumptions are then based on expert knowledge and intuition of the modeller. In this case, our modelling approach mainly supports in ensuring consistency within this model of thought. Secondly, the modelling process can also integrate existing knowledge about the product lifecycle which is based on statistics and experiences from previous products. The modelled states and especially the probabilistic transitions therefore either stem from underlying beliefs and assumptions or existing knowledge of previous products and their lifecycles.

The concept of hierarchy and nested states simplifies and structures the process of modelling a complex lifecycle. The modeller can start by modelling the lifecycle on a very high level - for example starting with three states only: production, use and end of life. He can then further define the processes underlying theses high level states, adding child states and connecting them via transitions on a lower level. Thereby, the degree of detail in the lifecycle model is up to the modeller.

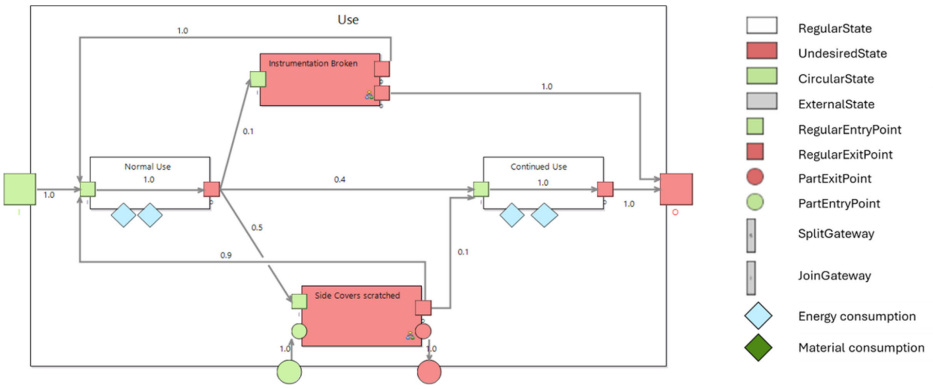


Figure 4. Process diagram representing the sequence of the child states within a parent state

We developed several visualization views and tools to simplify the modelling process and reduce complexity. The main modelling view - the Process Diagram - addresses the sequential relationship of states (see Figure 4). To lower complexity, it depicts the lifecycle on only two levels of hierarchy at once: A state and its directly contained child states with all transitions. The modeller can use this diagram to

https://doi.org/10.1017/pds.2026.10644 Published online by Cambridge University Press

review the lifecycle, alter the sequence of states and identify critical points in the lifecycle. A second view provides the user with an overview of the state's hierarchy in a tree structure (see Figure 5). This view does not provide any information on the sequential order of states.

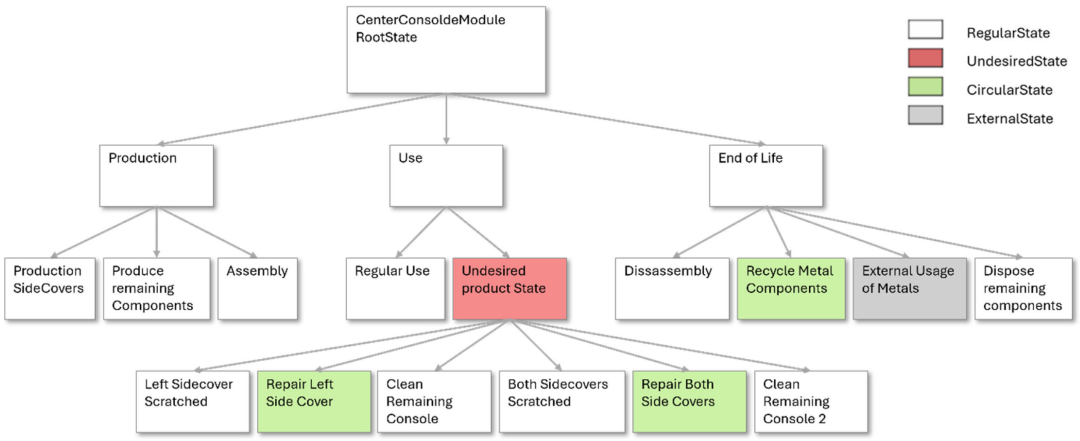


Figure 5. Hierarchical state structure with parent states and their contained child states

4. Implementation and evaluation

The hierarchical lifecycle model was employed and evaluated during the development of a circular automotive center console. It was implemented in an MBSE software tool for circular product design, developed within the Cyclometric research project (Fraunhofer IAO, 2025). The project team consisted of five designers and engineers from an automotive manufacturer and a design agency. The center console development was a greenfield effort (see Figure 6). Because the tool was new, the designers' and engineers' interaction with it was mediated: an experienced operator handled the software, while the designers and engineers provided and interpreted all lifecycle and product related inputs and outputs. It should be noted that the purpose of this evaluation is to demonstrate the applicability and practical feasibility of the proposed hierarchical lifecycle model rather than to quantitatively validate improvements in circularity or sustainability performance. The evaluation therefore focuses on assessing whether the model can be consistently applied in a product development context and whether it adequately supports the modelling and reasoning required for circular product design decisions. Evaluation was conducted through a combination of external observation of impacts on the product development process and formal reasoning. We focused on the three challenges, described in Section 1.

1. Correct modelling of circularity measures and their implications for components: In total, the designers and engineers modelled 26 circularity measures to be applied to different components of the center console in the lifecycle - including repair, remanufacture, reuse, repurpose, and recycle. We were able to represent all these measures and assign them correctly to modules and components at different hierarchy levels. Furthermore, we captured the implications of these measures for child components. For example, remanufacturing the armrest of the center console required removing, repairing and washing the textiles. Consequently, our model enables correct modelling of circularity measures and their implications for child components.
2. Consistent modelling of lifecycles across hierarchies: This feature is inherently guaranteed by the metamodel. States of child components can only be modeled within states of their parent components. This prevents child components from being in states that conflict with those of their parent - except when they are disassembled and follow a different lifecycle from their parent component. For example, the side cover is replaced and either recycled or reused in a different car, if it becomes scratched during production or logistics. Split and join gateways, as well as part-entry and part-exit points, ensure that such lifecycle branching occurs only at predefined points, while also ensuring that the parent component receives a replacement component from another source. Thus, this challenge is solved formally by the model's design.
3. Ineffective application of circular measures: A formal proof that using the hierarchical model leads to a more circular product design cannot be provided, because a reference case (a center-console design without the model's support) is missing. Nevertheless, the engineers and

https://doi.org/10.1017/pds.2026.10644 Published online by Cambridge University Press

designers incorporated 26 circular measures into the center console's design (see above). Initially, they proposed a set of more than 50 measures arbitrarily assigned to high-level components or modules, without verifying whether the hierarchy level was appropriate from an ecological perspective or semantically correct (e.g., whether the intent was to remanufacture the whole module or only certain parts). Our hierarchical model required specificity. The team therefore reworked and reevaluated their circularity measure assignments and arrived at 26 final, well-specified solutions. The touch sensitive textile, for example, had eleven circular measures assigned to it in the first step, which was broken down to three component-specific ones in the end. Consequently, the model encouraged a more critical and precise application of circularity measures, which we argue leads to better employment of circular design principles in general.

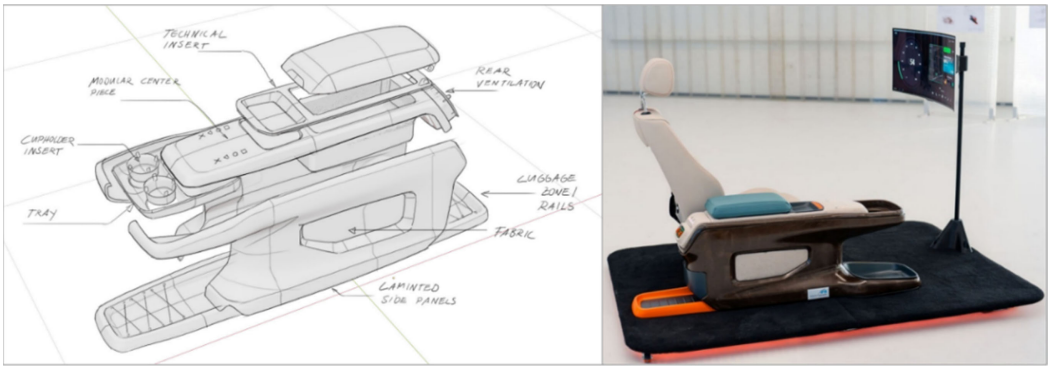


Figure 6. Developed circular automotive center console (Fraunhofer, 2025)

5. Discussion and conclusion

This paper introduced a hierarchical probabilistic lifecycle model for circular product development using stochastic petri nets. By defining states with explicit contexts, nested hierarchies, and split/join gateways, the model enables component-specific circularity measures while maintaining formal consistency between parent and child components. In doing so, it addresses three persistent challenges: correct representation of circularity measures and their implications for components, consistent modelling across hierarchy levels, and targeted, component-appropriate application of circularity measures. The evaluation in the development of a circular automotive center console demonstrated these benefits: the hierarchical model not only provides a means to represent lifecycles for hierarchical product architectures, but also ensures consistency, enables correct modelling of circularity measures, and supports the targeted application of circular design principles. Nonetheless, the approach also reduces complexity by structuring assumptions and decisions in a hierarchical manner and provides a robust foundation for early-stage circular design and subsequent LCA or trade-off analyses.

The current modelling approach still includes several simplifications: lifecycle state durations are modelled discretely, and the timing of probabilistic state transitions cannot be modelled continuously over time. This limits the precise scheduling of circularity measures, as the integration of those measures cannot be perfectly aligned with actual product or component degradation. Moreover, the model does not include an approach for history tracking, so the influence of earlier lifecycle stages on later behavior cannot be accounted for. Future work should therefore introduce time- and history-dependent probabilistic modelling to better represent continuous degradation, identify the optimal timing for the integration of circularity measures and consider effect of lifecycle events on later behaviour. Comprehensive validation - both regarding applicability in practice and quantitative benefits - is also required, along with broader industrial evaluation to assess scalability, integration into development processes, and opportunities for partial automation of the modelling process. Additionally, we aim to further investigate how LCA methods can quantify the impact of design choices on the overall product circularity and sustainability using the proposed lifecycle modelling approach.

Acknowledgement

This work was supported by a Fraunhofer ICON grant.

https://doi.org/10.1017/pds.2026.10644 Published online by Cambridge University Press

References

Adamowicz, M. (2022). Green Deal, Green Growth and Green Economy as a Means of Support for Attaining the Sustainable Development Goals. *Sustainability*, 14(10), 5901. https://doi.org/10.3390/su14105901

Anand, A., & Wani, M. F. (2010). Product Life-Cycle Modeling and Evaluation at the Conceptual Design Stage: A Digraph and Matrix Approach. *Journal of Mechanical Design*, 132(9), Article 091010. https://doi.org/10.1115/1.4002241

Bhander, G. S., Hauschild, M., & McAloone, T. (2003). Implementing life cycle assessment in product development. *Environmental Progress*, 22(4), 255–267. https://doi.org/10.1002/ep.670220414

Block, L., Werner, M., Spindler, H., & Schneider, B. (2023). A Variability Model for Individual Life Cycle Paths in Life Cycle Engineering. *Future Automotive Production Conference 2022*, 73–85. https://doi.org/10.1007/978-3-658-39928-3_6

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

Dér, A., Kaluza, A., Reimer, L., Herrmann, C., & Thiede, S. (2022). Integration of Energy Oriented Manufacturing Simulation into the Life Cycle Evaluation of Lightweight Body Parts. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 9(3), 899–918. https://doi.org/10.1007/s40684-021-00412-w

Ellen MacArthur Foundation. (2013). *Towards the Circular Economy: Economic and business rationale for an accelerated transition*. https://ellenmacarthurfoundation.org/publications

Fraunhofer (Ed.). (2025). Cyclometric: - Software und Methoden für die nachhaltige und kreislauforientierte Entwicklung in der frühen Konzeptphase. *Abschlussbericht.* https://doi.org/10.24406/publica-5589

Fraunhofer IAO. (2025). *CYCLOMETRIC - Nachhaltige Produktentwicklung neu gedacht*. https://cyclometric.de/

Halstenberg, F. A., Lindow, K., & Stark, R. (2019). Leveraging Circular Economy through a Methodology for Smart Service Systems Engineering. *Sustainability*, 11(13), 3517. https://doi.org/10.3390/su11133517

Jeswiet, J. (2014). Life Cycle Engineering. In CIRP Encyclopedia of *Production Engineering* (pp. 757–758). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-20617-7_6609

José Potting, M.P., Hekkert, E. Worrell, & Aldert Hanemaaijer (2017). Circular Economy: Measuring Innovation in the Product Chain. https://research-portal.uu.nl/en/publications/circular-economy-measuring-innovation-in-the-product-chain

Sanford Friedenthal (2008). *A Practical Guide to SysML: The Systems Modeling Language*. https://www.academia.edu/62142055/A_Practical_Guide_to_SysML_The_Systems_Modeling_Language

Schwahn, M., Potinecke, T., Block, L., Werner, M. J., & Tarlosy, F. S. (2024). Enabling the design for circularity through circularity measures: Breaking down the R-strategies into useful design measures. *Proceedings of the Design Society*, 4, 2745–2754. https://doi.org/10.1017/pds.2024.277

Shu, Q., & Wang, C. (2007). A conceptual framework for product lifecycle modelling. *Enterprise Information Systems*, 1(3), 353–363. https://doi.org/10.1080/17517570701515233

Umeda, Y., Fukushige, S., Kunii, E., & Matsuyama, Y. (2012). LC-CAD: A CAD system for life cycle design. *CIRP Annals*, 61(1), 175–178. https://doi.org/10.1016/j.cirp.2012.03.043

Wouterszoon Jansen, B., van Stijn, A., Gruis, V., & van Bortel, G. (2020). A circular economy life cycle costing model (CE-LCC) for building components. *Resources, Conservation and Recycling*, 161, 104857. https://doi.org/10.1016/j.resconrec.2020.104857

Yvars, P.A., & Zimmer, L. (2021). A Model-based Synthesis approach to system design correct by construction under environmental impact requirements. *Procedia CIRP*, 103, 85–90. https://doi.org/10.1016/j.procir.2021.10.013

https://doi.org/10.1017/pds.2026.10644 Published online by Cambridge University Press