Enabling Informed Sustainability Decisions: Sustainability Assessment in Iterative System Modeling

Gabriele Gramelsberger
Theory of Science and Technology
RWTH Aachen University
gramelsberger@humtec.rwth-aachen.de

Hendrik Kausch
Software Engineering
RWTH Aachen University
kausch@se-rwth.de

Judith Michael [*]
Software Engineering
RWTH Aachen University
michael@se-rwth.de

Frank Piller
Technology and Innovation Management
RWTH Aachen University
piller@time.rwth-aachen.de

Ferdinanda Ponci
E.ON Research Center
at RWTH Aachen University
fponci@eonerc.rwth-aachen.de

Aaron Praktiknjo
E.ON Research Center
at RWTH Aachen University
apraktiknjo@eonerc.rwth-aachen.de

Bernhard Rumpe
Software Engineering
RWTH Aachen University
rumpe@se-rwth.de

Rega Sota
School of Business and Economics
RWTH Aachen University
rega.sota@socecon.rwth-aachen.de

Sandra Venghaus
School of Business and Economics
RWTH Aachen University
venghaus@socecon.rwth-aachen.de

***Abstract*—When planning, creating, and evolving systems throughout their lifecycle, it is essential to assess their impact on our world. Despite this pressing need, existing structured methods for systematically assessing social, economic, and environmental impacts are not related to targets of the United Nations' sustainable development goals. Moreover, existing Architecture Description Languages (ADLs) lack concepts and tooling for sustainability assessment. Our aim is to allow modeling systems, their sustainability properties, and sustainability questions in a structured manner for a wide domain. This paper proposes the engineering and design of a domain-specific language for sustainability assessment embedded into ADLs and showcases its use for evaluating a citizen energy community system as a case study. We discuss possibilities on how to use such models in their further processing and explore challenges in technical realization. This initial step towards standardizing the sustainability assessments of modeled systems throughout the development is both comprehensive and formal so that developers can make informed, sustainable decisions based on consequence assessments upfront.**

***Index Terms*—Systems Engineering, Domain-Specific Languages, Model-Driven Engineering, Sustainable Development Goals, Life-Cycle Sustainability Assessment, Architecture Description Language, Energy Planning**

I. Introduction

Motivation. When developing and evolving systems, technologies, and processes over a longer period of time sustainability plays a significant role in each decision point of developers. Such systems include the production domain, Internet of Things (IoT), Cyber-Physical System (CPS), or pure software systems. Development decisions may lead to negative influences on different sustainability goals [1] in the areas of social, economic, and environmental sustainability [2].

[*]Corresponding author.

Research Question. We tackle the main research question of how to enable system developers to iteratively evolve a system throughout its life cycle in a sustainable way.

Contribution. To make these informed decisions, we suggest a model-based approach that incorporates sustainability descriptions in Architecture Description Language (ADL) models. This paper explores and introduces an approach to combine sustainability assessment defined in a Domain-Specific Language (DSL) with ADL models throughout an iterative development process. The approach supports decision-making through a system's evolution and leads to the implementation of sustainable systems. As a running example, we show two development scenarios in an energy planning case study for a citizen energy community.

Structure. We provide foundations, use a running example to introduce the methodology for sustainable system development, and conclude with a roadmap for implementing DSLs and assessing sustainability.

II. Preliminaries

Architecture Description Languages. For modeling systems, ADLs [3] offer great possibilities for iterative development. Most ADLs follow the component-connector approach, where a system architecture is defined by its components/parts and their connectors/ports. Often, additional (behavior) description possibilities are offered for atomic components through language compositions, e.g., state charts [4]. Components define their communication interface through input and output ports.

Composed systems use components and define connections between their ports. This modular approach permits introducing different hierarchical levels to facilitate necessary abstraction and readability. For the *evolution of system models*, ADLs offer great options: By clear modularisation of component interfaces, hierarchical levels with sub-components, and component behavior, the system engineers can adapt and evolve relevant parts. Necessary implementation details can be added later on while maintaining great abstraction for the early development process, e.g., by defining a high-level abstract system view while avoiding implementation details, by introducing new hierarchical levels, and by decomposing functions into smaller sub-functions throughout the system's evolution.

Sustainability. The United Nations considers sustainability one of its top priorities and has developed several programs to promote sustainable development worldwide. Sustainability refers to the *ability to meet the needs of the present generation without compromising the ability of future generations* to meet their own needs. It is about the responsible use of resources, protecting the environment, and promoting social justice. The 2030 Agenda for Sustainable Development with its 17 Sustainable Development Goals (SDGs)[1] is a central framework for the UN's global commitment to sustainability. Each UN country aggregates and provides sustainability indicator values [5]. Indicators influence different SDGs positively or negatively and are used to measure SDG targets. Interlinkages between SDG targets are still under research, c.f. researched interlinkages between targets provided by the EU [6].

Sustainability Assessment. Sustainable development requires the consideration of environmental, social, and economic aspects [2]. Sustainability assessments evaluate and analyze these aspects to ensure that decisions are made following sustainability principles. Iterative sustainability assessments enable companies, policymakers, and organizations to measure and track their impacts. This helps them improve their performance by implementing more sustainable practices and identifying potential problems throughout development. Common practices providing assessment methods such as life-cycle thinking [7], Life-Cycle Assessment (LCA) [8], or Life-Cycle Sustainability Assessment (LCSA) [9] provide a set of indicators related to different areas of sustainability.

III. Approach

We present our methodology that combines a sustainability DSL with an ADL to enable informed development decisions in iterative system development processes. The assessment process is explored using an *energy planning* running example for different *citizen energy community* [10] aspects.

Citizen energy communities (see Figure 1) comprise citizens and small commercial entities, with local energy generation and storage capacity. The community facilitates the possibility to purchase and sell energy and energy services to optimize local consumption and enable local energy trading. The citizens actively interact with the electrical distribution

[1]For additional information about SDGs see https://sdgs.un.org/goals

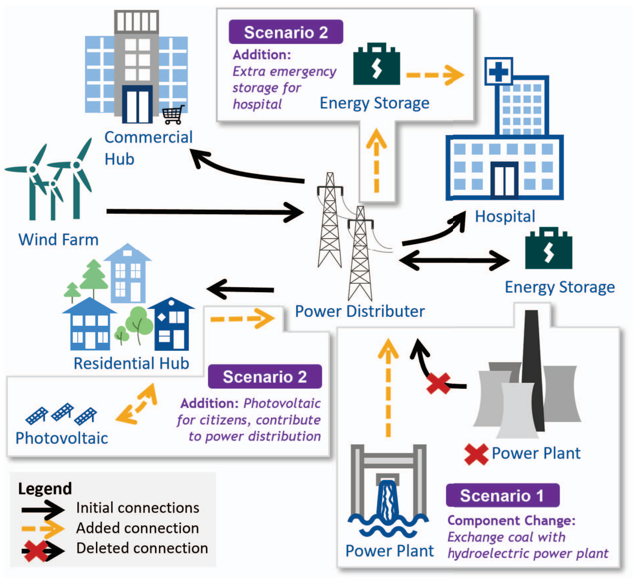

Fig. 1. Citizen energy community, architecture for system evolution. Scenario 1: evolution of electricity generator by exchanging a coal power plant with a hydroelectric power plant. Scenario 2: architectural evolution and integration of photovoltaic systems for citizens.

system and its stakeholders according to various regulatory models. The sustainability of such communities depends on, e.g., the technologies for power components and automation, how the distribution system operator is or is not reimbursed for grid usage, and how the community impacts the landscape. Additionally, scalability, location, and to what extent energy communities generate jobs and can be operated, interplay with various other aspects for the sustainability assessment. Their interactions and combinations of these factors within a holistic model of the energy community are not explicit. Thus sustainability quantification and analysis are practically impossible without a comprehensive ground model in a suitable DSL and the possibility to interface domain-specific applications.

Using already existing ADLs and combining them with a sustainability DSL through language composition is a first step towards continuous sustainability assessment throughout the development and evolution of systems, as they provide the necessary level of abstraction. Language workbenches allow for easy composition and rich functionality [11], [12]. Developing a system's functional architecture starts with abstract ADL models using, e.g., SysML [13], MontiArc [14], or UML+ROOM [15]. These models are further refined adding details throughout development.

In this paper, we take a closer look at two development scenarios. In Scenario 1, a component is (ex)changed, while leaving the overall system intact. In Scenario 2, we are introducing new components and changing connections. These two scenarios are introduced and then intertwined with sustainability description and assessment options to allow continuous sustainability assessment. The changes to the system can be divided into simple development patterns (see [16]).

```
 1 package citizenenergycommunity;
 2 import library.energyplanning;
 3
 4 component CitizenEnergyCommunity{
 5   autoconnect;
 6   port
 7     ...
 8
 9   component Hospital hospital;
10   component CommercialHub comHub;
11   component ResidentialHub resHub;
12   component WindFarm windfarm;
13   component PowerDistributor distrib;
14   component EnergyStorage storage;
15   component CoalPowerplant powerplant;
16
17   satisfy sustainability{
18     sdg: [7,11,13]
19     ...
20   }
21 }
```

Listing 1. Initial citizen energy community model (excerpt)

In Listing 1, the structure of a citizen energy community is modeled in MontiArc defining components, ports, and connections. Through the package system, components and their interfaces from other models can be included in the citizen energy community model (Listing 1, l.1). The sustainability DSL is defined for SDG indicators, but extended by an DSL library incorporating domain-specific indicators for the energy planning sector (l.2). The complete system is defined as a component to facilitate the hierarchical modeling of systems (l.4). Components with identical port names are automatically combined using the keyword `autoconnect` (l.5). For this system, we realize the connections from Figure 1. The sub-components of the system are instantiated (l.9-15). Finally, sustainability goals are defined as constraints the system must satisfy (l.17). Here, the SDGs 7 (affordable & clean energy), 11 (sustainable cities & communities), and 13 (climate action) are chosen for the assessment (l.18).

```
 1 component CoalPowerplant{
 2   port
 3     out ElectricalEnergy ee;
 4
 5   sustainability{
 6     type: energy, structure, process;
 7     indicators{
 8       consumption: coal;
 9       co2Emission: 950 gCO2/kWh;
10       landscapeUsage: 1km^2;
11       ...
12     }
13   }
14 }
```

Listing 2. Coal power plant model (excerpt)

Each of the components is extended by a sustainability DSL model describing its sustainability indicators. A `CoalPowerplant`'s interface (Listing 2, l. 2-3) has one output port providing electrical energy. Additionally to the component types (l. 6), we define sustainability indicators, e.g., for consumption types, CO_2 emissions, and landscape usage (l. 7-10). Using this sustainability description for all components, the modeled sustainability goal fulfillment for the system can be assessed by experts or passed on to assessment systems.

Process Step Sustainability Assessment.
When specifying the sustainability of systems one must pay attention to different aspects: Some components might be built from carbon-heavy concrete and are unsustainable from a production perspective, while other components might be unsustainable by producing named concrete. The sustainability of a whole system is not straightforwardly defined by its component's sustainability. Maybe the components built from carbon-heavy concrete already exist and the component producing named concrete is used so sparingly, that the overall benefit overweights its costs. Furthermore, sustainability is not purely environmental, but an interplay between social, environmental, and economic aspects. In this paper, we explore an idea to define sustainability indicators [17] for atomic components individually, but inferring systems' sustainability indicators automatically. Furthermore, a system's sustainability targets can be defined in the model. Then, experts (real persons), a tool, or structures like the envisioned SEER (see [18]) assess the system from the aggregated information.

In our example, after an initial assessment of the planned citizen energy community, the assessment showed alarming levels of CO_2 emissions with a negative impact on SDG 13 (climate action), only minor benefits for SDG 7 (affordable & clean energy), and no positive influence on SDG 11 (sustainable cities & communities). As the negative impact on SDG 13 is most problematical, the sub-component's influence is analyzed, and the coal power plant is identified as the main contributor. The system is then changed.

Scenario 1: Changing Components
Different options arise when (ex)changing a single component while maintaining the overall system functionality: (1) A component's functional behavior can be refined by adding details, thus, removing underspecification. (2) A component's functionality can be decomposed into smaller subfunctions. This creates a new hierarchical level in the system, but the black-box architecture is unchanged. Further developing a system by iteratively applying these patterns offers higher abstraction in the beginning and concrete definitions later on.

```
 1 component HydroPowerplant{
 2   port
 3     out ElectricalEnergy ee;
 4
 5   sustainability{
 6     type: energy, structure, process;
 7     indicators{
 8       consumption: renewable, hydro;
 9       co2Emission: 24 gCO2/kWh;
10       landscapeUsage: 2km^2;
11       ...
12     }
13   }
14 }
```

Listing 3. Hydroelectric power plant model (excerpt) replaces coal power plant model (Listing 2)

In our example, the coal power plant is exchanged with a hydroelectric power plant (see Figure 1 and Listing 3).

Assessment. The replacement of a polluting coal power plant changed the components' sustainability properties, e.g., reduced the power plant's CO_2 emissions by over 95%. Using the new sustainability indicators from Listing 3, a new sustainability assessment can be done. The results now show slightly positive effects for SDGs 7, 11, and 13.

Scenario 2: Changing Architecture

When developing systems, architectural changes on the same hierarchical level often occur. These changes can introduce or remove abstraction, by (de)composing components, change the connections between components, and add or delete whole components including connections.

By adding photovoltaic units to the residential hub (see Figure 1 and l. 3 in Listing 4), the residents are directly provided with electric energy, and any surplus is directly fed into the distribution grid and sold to the power retailer. This further enhances the original vision of the citizen energy community, where citizens play an active part in the energy market. Furthermore, the hospital is connected with its emergency energy storage that provides energy even in case of general power outages (see Figure 1 and l. 4 in Listing 4). For this, the already existing component `EnergyStorage` is reused by instancing it twice in this architecture, as the general `storage` (Listing 4. l. 1) and as the emergency storage `emStorage` (l. 4).

```
1  component EnergyStorage storage;
2  component HydroPowerplant powerplant;
3  component Photovoltaic photo;
4  component EnergyStorage emStorage;
5
6  satisfy sustainability{
7    sdg: [7,11,13]
8    ...
9  }
```

Listing 4. Advanced citizen energy community model (excerpt), start from line 14 in Listing 1

Assessment. The system evolution introduced a new photovoltaic component changing the energy infrastructure (see Figure 1, Scenario 2). This, in turn, changed the system's sustainability properties. The previously assessed SDGs 7, 11, and 13 are assessed again, and the iteratively derived architecture shows promising results for all assessed SDGs. To maintain the currently modeled system's functionality but still iteratively evolve it, refactoring techniques [19] can be used. By integrating continuous sustainability assessment early on in the planning of a new citizen energy community, decision-makers were directed toward sustainable decisions.

IV. Roadmap

To accomplish the vision of continuous, integrated, and automated sustainability assessment throughout system development and evolution, we identified several needed steps.

Refining and evaluating a stable sustainability DSL which is *easily extendable* for different domains is important. While the case study shows promise in enabling informed sustainability decisions, there are also some limitations from the modeler's perspective. The energy planning domain expert is able to define sustainability indicators and their values for components in a familiar way using domain-specific indicators (Listing 2, l. 7-10), but the expert might not always be familiar with the SDGs or the relevant evaluation criteria of the overall system (e.g, Listing 1, l. 18). To overcome this, we need to provide tooling or low-code UIs [20] with additional information for SDGs or types. The combination with *model-driven engineering of digital twins for systems* [21] should be evaluated since this may offer an easy way to assess and improve already existing (cyber-physical) systems [22]–[24], their engineering [25], or complex software systems and their development processes [26], [27]. *Verification of sustainability constraints* can be checked by combining the monitoring capabilities of digital twins with sustainability constraint models analogous to [28]. Furthermore, *delta modeling* [29], [30] could be integrated to offer simple and effective *evolution* options to system engineers. Clear and general *interfaces* for future extensions play a major role. On the one hand, suppliers that evaluated their components should be able to provide the *assessment results* to the buyer, who in turn integrates it easily into their architecture. On the other hand, *adding domain-specific libraries* for the DSL is paramount. Most domains have specific indicators, e.g., LCSA indicators, but their *interlinkages to SDG indicators and goals are not well researched* and might even be country or region-dependent [31]. As a global research community, these gaps can gradually be closed, and new research results should then easily be integrated into the sustainability DSL.

Additionally, *metrics must be evaluated*, and the compositionality of indicators must be researched. When aggregating indicators for a composed system, e.g., Listing 4, it might be easy to just add up the CO_2 emission of all subcomponents, but some indicators have other compositional properties. The SDG indicator 11.7.1 "Average share of the built-up area of cities that is open space for public use for all, by sex, age and persons with disabilities" is influenced differently by the landscape usage of industry areas and parks. Even in between parks, a type distinction is necessary, e.g., a wildlife park might contribute positively to the indicator, while a skate park without accessibility for persons with disabilities does not.

Model-processing tooling must be provided to automate sustainability assessment where possible. For this, interfaces for existing tooling, e.g., [32]–[36], must be defined or new tooling developed. Ideally, *easy extendability for different sustainability assessment tools* must be facilitated and assisted.

Once this process is generally applicable, it must be further *integrated into society* to fully leverage the sustainability benefits. One possibility in this regard is the introduction of a *certificate for sustainable system development*. This certificate can either be voluntary or mandatory. In the case of a *voluntary certificate*, success is dependent on consumer behavior. If the consumers value sustainability highly and actively choose products or companies proving sustainability, a sustainability certificate should provide an economic edge and entice companies to sustainable behavior. Additionally, even in business-to-business areas, the certificate could guarantee continuously sustainable supply chains. Alternatively, introducing a *mandatory certificate* might provide faster effects. An example of successful mandatory integration is avionic certificates, e.g., for airplanes. By also providing guidelines for attaining the certificates [37], the industry must rigorously test, validate, and verify safety-critical systems.

Besides ADLs, there are *different approaches and kinds of models describing systems that might profit from sustainability specifications and assessments*, e.g., feature models for software product line evolutions [38], [39]. By embedding the sustainability DSL in modeling languages used for *reference architectures* and verifying the sustainability of the reference architecture, we predict reduced effort for the development of sustainable incarnations of reference architectures when combined with conformance checks [40].

In conclusion, this paper presents an approach to facilitate informed sustainability decision-making throughout the lifecycle of systems during their development. Our citizen energy community case study shows that to reach full maturity, an open science approach with researchers from different disciplines is needed. An open science approach enables global researchers to extend core functionalities, e.g., to offer sustainability assessments, leading to informed decisions across various domains, industries, and locations around the world.

References

[1] B. K. Sovacool, P. Newell, S. Carley, and J. Fanzo, "Equity, technological innovation and sustainable behaviour in a low-carbon future," *Nature human behaviour*, vol. 6, no. 3, pp. 326–337, 2022.

[2] B. Purvis, Y. Mao, and D. Robinson, "Three pillars of sustainability: in search of conceptual origins," *Sust. Science*, vol. 14, no. 3, 2019.

[3] N. Medvidovic and R. N. Taylor, "A Classification and Comparison Framework for Software Architecture Description Languages," *IEEE Transactions on software engineering*, vol. 26, no. 1, pp. 70–93, 2000.

[4] A. Butting, O. Kautz, B. Rumpe, and A. Wortmann, "Architectural Programming with MontiArcAutomaton," in *12th Int. Conf. on Software Engineering Advances (ICSEA)*. IARIA XPS Press, May 2017.

[5] U. Nations, "SDG Global Database," 2021. [Online]. Available: https://unstats.un.org/sdgs/dataportal

[6] "SDG interlinkages visualization tool - Target level," Joint Research Centre of the European Commission (JRC-EC). [Online]. Available: https://knowsdgs.jrc.ec.europa.eu/interlinkages-goals

[7] A. Mazzi, "Chapter 1 - Introduction. Life cycle thinking," in *Life Cycle Sustainability Assessment for Decision-Making*. Elsevier, 2020.

[8] A. Tukker, "Life cycle assessment as a tool in environmental impact assessment," *Environmental Impact Assessment Review*, vol. 20, no. 4, pp. 435–456, 2000.

[9] M. Finkbeiner, E. M. Schau, A. Lehmann, and M. Traverso, "Towards Life Cycle Sustainability Assessment," *Sustainability*, vol. 2, 2010.

[10] EU, "Directive (EU) 2018/2001 of the European Parliament and of the Council of 11 December 2018 on the promotion of the use of energy from renewable sources (recast)," *Off. Journal of the EU*, 2018.

[11] K. Hölldobler, O. Kautz, and B. Rumpe, *MontiCore Language Workbench and Library Handbook: Edition 2021*, ser. Aachener Informatik-Berichte, Software Engineering, Band 48. Shaker Verlag, May 2021.

[12] A. Haber, M. Look, P. Mir Seyed Nazari, A. Navarro Perez, B. Rumpe, S. Völkel, and A. Wortmann, "Composition of Heterogeneous Modeling Languages," in *Model-Driven Engineering and Software Development*, ser. CCIS, vol. 580. Springer, 2015.

[13] Object Management Group, "Systems Modeling Language V. 1.6," 2023. [Online]. Available: https://www.omg.org/spec/SysML/1.6

[14] A. Haber, J. O. Ringert, and B. Rumpe, "MontiArc - Architectural Modeling of Interactive Distributed and Cyber-Physical Systems," RWTH Aachen University, Technical Report AIB-2012-03, February 2012.

[15] B. Rumpe, M. Schoenmakers, A. Radermacher, and A. Schürr, "UML + ROOM as a Standard ADL?" in *Engineering of Complex Computer Systems (ICECCS'99)*, ser. IEEE, 1999.

[16] J. Philipps and B. Rumpe, "Refinement of Information Flow Architectures," in *ICFEM'97*. IEEE, 1997.

[17] U. Nations, "SDG Indicators," 2023. [Online]. Available: https://shorturl.at/muvD4

[18] J. Kienzle and et al., "Toward model-driven sustainability evaluation," *Communications of the ACM*, vol. 63, no. 3, pp. 80–91, 2020.

[19] J. Philipps and B. Rumpe, "Refactoring of Programs and Specifications," in *Practical Foundations of Business and System Specifications*. Kluwer Academic Publishers, 2003, pp. 281–297.

[20] M. Dalibor, M. Heithoff, J. Michael, L. Netz, J. Pfeiffer, B. Rumpe, S. Varga, and A. Wortmann, "Generating Customized Low-Code Development Platforms for Digital Twins," *Journal of Computer Languages (COLA)*, vol. 70, 2022.

[21] S. Fur, M. Heithoff, J. Michael, L. Netz, J. Pfeiffer, B. Rumpe, and A. Wortmann, *Sustainable Digital Twin Engineering for the Internet of Production*. Springer Nature Singapore, 2023, pp. 101–121.

[22] J. C. Kirchhof, J. Michael, B. Rumpe, S. Varga, and A. Wortmann, "Model-driven Digital Twin Construction: Synthesizing the Integration of Cyber-Physical Systems with Their Information Systems," in *Int. Conf. on Model Driven Engineering Languages and Systems*, 2020.

[23] P. Bibow, M. Dalibor, C. Hopmann, B. Mainz, B. Rumpe, D. Schmalzing, M. Schmitz, and A. Wortmann, "Model-Driven Development of a Digital Twin for Injection Molding," in *Int. Conf. on Advanced Information Systems Engineering (CAiSE'20)*, 2020.

[24] D. Bano, J. Michael, B. Rumpe, S. Varga, and M. Weske, "Process-Aware Digital Twin Cockpit Synthesis from Event Logs," *Journal of Computer Languages (COLA)*, vol. 70, June 2022.

[25] J. Michael, I. Nachmann, L. Netz, B. Rumpe, and S. Stüber, "Generating Digital Twin Cockpits for Parameter Management in the Engineering of Wind Turbines," in *Modellierung 2022*. GI, June 2022, pp. 33–48.

[26] M. Heithoff, A. Hellwig, J. Michael, and B. Rumpe, "Digital Twins for Sustainable Software Systems," in *GREENS 2023*. IEEE, 2023.

[27] T. Yarally, L. Cruz, D. Feitosa, J. Sallou, and A. van Deursen, "Uncovering Energy-Efficient Practices in Deep Learning Training: Preliminary Steps Towards Green AI," in *2023 IEEE/ACM 2nd Int. Conf. on AI Engineering – Software Engineering for AI (CAIN)*, 2023, pp. 25–36.

[28] T. Kurpick, M. Look, C. Pinkernell, and B. Rumpe, "Modeling Cyber-Physical Systems: Model-Driven Specification of Energy Efficient Buildings," in *Modelling of the Physical World (MOTPW'12)*. ACM, 2012.

[29] A. Haber, T. Kutz, H. Rendel, B. Rumpe, and I. Schaefer, "Delta-oriented Architectural Variability Using MontiCore," in *Software Architecture Conference (ECSA'11)*. ACM, 2011, pp. 6:1–6:10.

[30] A. Haber, K. Hölldobler, C. Kolassa, M. Look, K. Müller, B. Rumpe, and I. Schaefer, "Engineering Delta Modeling Languages," in *Software Product Line Conference (SPLC'13)*. ACM, 2013, pp. 22–31.

[31] J. G. Backes and M. Traverso, "Life cycle sustainability assessment as a metrics towards SDGs agenda 2030," *Current Opinion in Green and Sustainable Chemistry*, vol. 38, p. 100683, 2022.

[32] "openlca: The world's leading, high performance, open source life cycle assessment software." [Online]. Available: https://www.openlca.org/

[33] "Eco-efficiency software – umberto." [Online]. Available: https://www.ifu.com/umberto/

[34] "Sphera: Product sustainability software & data." [Online]. Available: https://sphera.com/product-sustainability-software/

[35] "Simapro: Lca software for informed change-makers." [Online]. Available: https://simapro.com/

[36] "Building for Environmental and Economic Sustainability (BEES)." [Online]. Available: https://www.wbdg.org/additional-resources/tools/bees

[37] T. K. Ferrell and U. D. Ferrell, "Rtca DO-178C/EUROCAE ED-12C," *Digital Avionics Handbook*, 2017.

[38] S. Urli, M. Blay-Fornarino, P. Collet, and S. Mosser, "Using composite feature models to support agile software product line evolution," in *6th Int. Workshop on Models and Evolution (ME '12)*. ACM, 2012.

[39] J. Schröpfer, F. Schwägerl, and B. Westfechtel, "Consistency Control for Model Versions in Evolving Model-Driven Software Product Lines," in *Int. Conf. on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. IEEE, 2019.

[40] A. Bucaioni, A. Di Salle, L. Iovino, I. Malavolta, and P. Pelliccione, "Reference Architectures Modelling and Compliance Checking," *Software and Systems Modeling*, vol. 22, no. 3, pp. 891–917, 2023.