



UNIVERSITÀ DI PARMA

UNIVERSITÀ DEGLI STUDI DI PARMA

DOTTORATO DI RICERCA IN
Ingegneria Industriale

CICLO XXXVIII

AI-Driven engineering product data structuring for LCA and foreground LCI modeling

Coordinatore:

Chiar.mo Prof. Alessandro Tasora

Tutore:

Chiar.mo Prof. Claudio Favi

Dottoranda: Sarra Brahem

Anni Accademici 2022/2023 – 2024/2025

Dottorati di Ricerca – Ph.D. Positions(DM 352)

Ente finanziatore: progetto finanziato dall'Unione Europea - NextGenerationEU – Piano Nazionale Ripresa e Resilienza (PNRR) - Missione 4 Componente 2 Investimento 3.3 “Introduzione di dottorati innovativi che rispondono ai fabbisogni di innovazione delle imprese e promuovono l'assunzione dei ricercatori dalle imprese” – Avviso N. 352 del 09/04/2022 del Ministero dell'Università e della Ricerca; Funder: Project funded under the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 3.3 - Call for tender No. 352 of 09/04/2022 of Italian Ministry of University and Research funded by the European Union – NextGenerationEU



ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Professor Claudio Favi, for his guidance, insight, and continued support throughout the course of this PhD at the University of Parma.

I would also like to thank my colleagues Núria Boix Rodríguez and Gabriel Araújo de Lima for their collaboration and support throughout this research. My thanks go to the companies that made this work possible, for providing the industrial data and access essential to the case studies presented in this thesis.

I am also grateful to the research team in France with whom I had the opportunity to work, and to Professor Peggy Zwolinski for her supervision and guidance during that period.

Finally, my deepest gratitude goes to my family. Their unwavering support, patience, and encouragement carried me through every stage of this journey, especially in its most demanding moments. This achievement would not have been possible without them, and I am profoundly grateful for everything they have given me along the way.

LIST OF CONTENTS

LIST OF ABBREVIATION	1
LIST OF FIGURES	3
LIST OF TABLES.....	4
ABSTRACT.....	5
1. INTRODUCTION.....	7
1.1 Background and Motivation	7
1.2 Problem Statement.....	8
1.3 Research Aim and Objectives.....	10
1.4 Research Questions.....	10
1.5 Scope of the Research and contribution.....	11
1.6 Thesis Organization	13
2. STATE OF THE ART	15
2.1 Life Cycle Assessment Methodology	15
2.2 The Life Cycle Inventory Phase	18
2.3 LCA Database and Background Data	21
2.4 Challenges of LCI Generation in Industrial Practices	22
2.4.1 General Challenges of LCI Generation.....	23
2.4.2 Operational and Practical Difficulties in Data Collection and Processing	24
2.5 Artificial Intelligence in Life Cycle Assessment	29
2.6 AI Techniques for Structured Information Extraction for LCI Generation	46
2.6.1 Schema-guided structured extraction from documents.....	46
2.6.2 The structured/hierarchy-aware gap	47
2.6.3 AI methods applied to LCI/BoM-adjacent problems.....	48
2.7 Synthesis and Research Gap	50
3. WORKFLOW OF A MANUAL LCI GENERATION IN INTERNAL PRACTICE.....	52
3.1.1 The manual workflow.....	52
3.1.2 Bills of Materials and CAD files as data sources.....	54
3.1.3 Limitations of BoM data for LCI purposes.....	56
3.1.4 Foreground–background matching	57
3.1.5 Synthesis and motivation for the proposed methodology	59
4. PROPOSED AI-DRIVEN METHODOLOGY FOR FOREGROUND LCI-ORIENTED REPRESENTATION.....	61

4.1 Overview of the proposed methodology	61
4.2 Stage I, Deterministic preprocessing and prompt preparation	65
4.2.1 BoM structure detection and preprocessing.....	65
4.2.2 Dynamic Schema Generation and Prompt Construction	73
4.3 Stage II, LLM-Based Extraction and Hierarchical Reconstruction	86
4.3.1 LLM Selection, Inference, and Extraction Procedure.....	86
4.3.2 Expected LLM Output and target representation.....	88
4.3.3 Error Conditions and Robustness	90
5. VALIDATION OF THE PROPOSED AI-ASSISTED METHODOLOGY	92
5.1 Introduction	92
5.2 Technical validation	92
5.2.1 Qualitative validation.....	94
5.2.2 Quantitative validation.....	97
5.2.3 Discussion of technical validation	103
5.3 Industrial case study validation: Methodology Application in a Real-World Context	106
5.3.1 Description of the industrial case study	107
5.3.2 Reference foreground inventory: preparation and manual disassembly	108
5.3.3 Application of the proposed methodology to the case-study BoM.....	110
5.3.4 Comparison between automated and manual results	112
5.3.5 From extracted structure to foreground LCI: a worked example	114
6. DISCUSSION, CONCLUSION AND FUTURE PERSPECTIVES.....	118
6.1 Overall Discussion.....	118
6.2 Answers to Research Questions.....	120
6.3 Scientific and Practical Contributions	122
6.3.1 Scientific Contributions	122
6.3.2 Methodological Contributions	123
6.3.3 Practical Implications	124
6.4 Limitations of the Proposed Methodology	124
6.5 Future Research Directions.....	127
BIBLIOGRAPHY	130

LIST OF ABBREVIATION

Artificial InAdaptive Neuro-Fuzzy Inference System	ANFIS
Application Programming Interface	API
Artificial Intelligence	AI
Artificial Neural Network	ANN
Bidirectional Encoder Representations from Transformers	BERT
Bill of Materials	BoM
Building Information Modeling	BIM
Comma-Separated Values	CSV
Computer-Aided Design	CAD
Computer-Aided Manufacturing	CAM
Computer-Aided Technologies (CAD/CAM/CAE)	CAx
Decision Tree	DT
Deep Learning	DL
Deutsches Institut für Normung (German Institute for Standardization)	DIN
Enterprise Application Integration	EAI
Enterprise Resource Planning	ERP
Environmental Building Materials	EBM
Environmental Product Declaration	EPD
European Reference Life Cycle Database	ELCD
European Union	EU
Extended Producer Responsibility	EPR
Extensible Markup Language	XML
F1-score (harmonic mean of Precision and Recall)	F1
False Negative	FN
False Positive	FP
Gaussian Process Regression	GPR
Generative Pre-trained Transformer	GPT
Genetic Algorithm	GA
Global LCA Data Access (network)	GLAD
Global Warming Potential	GWP
Gradient Boosting Machine	GBM
Greenhouse Gas	GHG

Health Product Declaration	HPD
Industry Foundation Classes	IFC
International Organization for Standardization	ISO
International Reference Life Cycle Data System	ILCD
JavaScript Object Notation	JSON
K-Nearest Neighbors	KNN
Knowledge Graph	KG
Language Model-based Document Information Extraction and Localization	LMDX
Large Language Model	LLM
Life Cycle Assessment	LCA
Life Cycle Impact Assessment	LCIA
Life Cycle Inventory	LCI
Linear Regression	LR
Machine Learning	ML
Manufacturing Execution System	MES
Multilayer Perceptron	MLP
Multiple Linear Regression	MLR
Ontology Alignment Evaluation Initiative	OAIE
Optical Character Recognition	OCR
Product Life Cycle Support (standard)	PLCS
Product Lifecycle Management	PLM
Random Forest	RF
Research Question	RQ
Resource Description Framework	RDF
Retrieval-Augmented Generation	RAG
Sentence-BERT	SBERT
Standard for the Exchange of Product model data	STEP
Structured Query Language	SQL
Support Vector Machine	SVM
Support Vector Regression	SVR
Term Frequency–Inverse Document Frequency	TF-IDF
True Positive	TP
U.S. Life Cycle Inventory (database)	USLCI
United Nations	UN
Web Ontology Language	OWL

LIST OF FIGURES

Figure 2.1. The four phases of the Life Cycle Assessment methodology and their interrelation (goal and scope, LCI, LCIA, interpretation)	16
Figure 4.1 Proposed AI-Driven Methodology for Foreground LCI-Oriented Representation	63
Figure 4.2 Structure Differences between Dynamic Entity Schema and Container Schema.....	76
Figure 5.1 Correspondence between the original BoM worksheet and the generated JSON output for BoM 5, illustrating metadata separation, header detection, and hierarchical structuring.	95
Figure 5.2 Correspondence between the original BoM worksheet and the generated JSON output for BoM 6, illustrating metadata handling and the flat structure produced for a BoM with no explicit hierarchy indicator.....	95
Figure 5.3 Worked example of the quantitative validation procedure applied to BoM 1: extraction validation (Panel A) and hierarchy validation (Panel B).....	102
Figure 5.4 Example of the packaging module and the identified components considered during manual inventory development.	109
Figure 5.5 Examples of the metallic components of the Body module, including zinc-coated panels.	110
Figure 5.6 Door module disassembly and the main components identified during manual inventory development.....	110
Figure 5.7 Portion of the original industrial BoM, illustrating the implicit representation of hierarchical relationships through level indicators and row ordering.	111

LIST OF TABLES

Table 3.1 BoM limitations encountered during manual LCI generation, and how each was addressed in practice	56
Table 4.1 Key Differences between Dynamic Entity Schema and Container Schema	76
Table 4.2 Functional Components of the LLM Extraction Prompt	84
Table 5.1 Characteristics of the technical evaluation datasets	94
Table 5.2 Qualitative validation results across the evaluated BoMs	96
Table 5.3 Quantitative validation results: Precision, Recall, F1-score, and Level Accuracy for extraction and hierarchy reconstruction across the evaluated BoMs	103
Table 5.4 Component-level comparison between the automated pipeline output and the manually verified Body, Door, and Packaging modules	114

ABSTRACT

Life Cycle Assessment (LCA) is widely used to evaluate the environmental performance of industrial products, yet the generation of the Life Cycle Inventory (LCI), the phase in which environmental impact ultimately originates, remains largely manual. Engineering documents such as Bills of Materials (BoMs) contain detailed information about product composition and hierarchy, but this information is created for manufacturing purposes, is structurally heterogeneous across companies and even across products from the same manufacturer, and encodes product hierarchy only implicitly, typically through a numeric Level attribute and row ordering. Interpreting this implicit structure before physical disassembly and material identification can proceed is a recurring, expert-dependent source of manual effort, as documented through the teams's own industrial practice in the household appliance sector.

This research work proposes and validates an AI-driven methodology that automatically transforms heterogeneous industrial BoMs into structured, hierarchical, machine-readable representations suitable for foreground LCI development. The methodology combines a deterministic preprocessing stage, which detects the BoM header and metadata regions and dynamically generates a document-specific schema from the attributes actually present, with a Large Language Model (LLM)-based stage that extracts component information and reconstructs parent–child relationships from the BoM's Level attribute according to explicit, prompt-embedded hierarchy rules.

The methodology is evaluated through a technical validation across six BoMs of varying structure and scale, combining qualitative stage-level checks with quantitative Precision, Recall, and F1-score measures for both attribute extraction and hierarchical reconstruction, and through an industrial case study on a combination oven–microwave, comparing the automated output

against a reference inventory obtained by manual disassembly. Results show that attribute-level extraction is consistently reliable across formats, while hierarchy reconstruction is strong under moderate scale and an explicit numeric signal but degrades under two identified conditions: large, deeply nested BoMs and BoMs lacking a hierarchy indicator altogether. Within the industrial case study, the automated output fully reproduced the verified structure of three product modules spanning up to five hierarchy levels. These findings demonstrate that AI can meaningfully reduce the manual interpretation effort preceding foreground LCI preparation, while clarifying the specific structural conditions under which its reliability currently holds and where expert oversight remains necessary.

1. INTRODUCTION

1.1 Background and Motivation

Global warming, resource depletion, and increasing environmental degradation have become defining challenges of the twenty-first century. Industrial production systems are among the principal contributors to these environmental pressures due to their intensive consumption of materials and energy, the complexity of manufacturing processes, and the globalization of supply chains. In response, governments, industries, and research institutions are increasingly adopting strategies aimed at reducing environmental impacts while maintaining economic competitiveness.

Several international initiatives have reinforced this transition towards more sustainable production systems. The United Nations Sustainable Development Goals (SDGs), adopted in 2015, establish a global framework for promoting responsible consumption, sustainable industrialization, and climate action. Likewise, the European Green Deal seeks to achieve climate neutrality by 2050 through significant transformations in industrial production, resource efficiency, and circular economy practices. Achieving these ambitious objectives requires reliable methodologies capable of quantifying the environmental performance of products and supporting informed decision-making throughout their life cycle.

Life Cycle Assessment (LCA) has become one of the most widely recognized methodologies for evaluating the environmental impacts of products, processes, and services from raw material extraction to end-of-life management. Owing to its comprehensive life cycle perspective, LCA supports engineering eco-design, environmental product declarations, regulatory compliance, and sustainability-oriented decision-making across numerous industrial sectors. However, despite its methodological maturity and widespread adoption, conducting an LCA remains a demanding task in

industrial practice, particularly when assessments must be performed rapidly on complex products characterized by large volumes of engineering data.

The primary challenge lies not in the LCA methodology itself, but in the generation of the Life Cycle Inventory (LCI), which requires transforming raw engineering information into structured environmental inventory data. Although modern industries generate vast quantities of digital product information throughout the product development process, these data are primarily created to support engineering, manufacturing, and business activities rather than environmental assessment. Consequently, substantial manual effort is still required before such information can be used within an LCA study.

Recent advances in Artificial Intelligence (AI), particularly in machine learning and Large Language Models (LLMs), offer new opportunities to automate knowledge extraction and data structuring from heterogeneous engineering documents. These developments have the potential to significantly reduce the time and expertise required to generate LCI data, thereby improving the efficiency, consistency, and scalability of environmental assessments in industrial contexts.

This research is motivated by the need to bridge the gap between existing engineering information and the environmental data required for LCA through the development of an AI-driven methodology for the automated interpretation and structuring of industrial engineering documentation in support of foreground LCI preparation.

1.2 Problem Statement

The adoption of digital engineering systems has considerably increased the amount of product information available within manufacturing companies. Engineering data are routinely generated and managed through Enterprise

Resource Planning (ERP) systems, Product Lifecycle Management (PLM) platforms, Computer-Aided Design (CAD) software, and, most importantly, Bills of Materials (BoMs), which describe the composition and hierarchical structure of products.

Despite this abundance of digital information, the generation of LCI data remains predominantly manual. Engineering documents contain detailed descriptions of product components, quantities, and assemblies, but they rarely provide the environmental information required for LCA. Consequently, LCA practitioners must manually interpret engineering documentation, identify materials and manufacturing processes, reconstruct product structures, and associate each component with suitable datasets available in LCA databases.

This workflow requires substantial domain expertise and represents one of the most time-consuming stages of an LCA study. Moreover, because the large amount of these activities rely on expert interpretation rather than standardized procedures, the resulting inventories may vary between practitioners, limiting reproducibility and making large-scale industrial application difficult.

Although companies already possess the information required to describe their products, automatically transforming heterogeneous Bills of Materials into structured, machine-readable representations suitable for foreground LCI development remains a significant technical challenge. Bridging this gap between engineering product documentation and environmental inventory modelling constitutes the central research problem addressed in this research work.

1.3 Research Aim and Objectives

The overall aim of this research is to develop and validate an AI-driven methodology capable of automatically transforming heterogeneous industrial Bills of Materials into structured representations that support foreground Life Cycle Inventory development.

To achieve this aim, the research work pursues the following objectives:

- Analyze current industrial practices for foreground LCI generation and identify the principal limitations of existing manual workflows
- Investigate the potential of AI techniques for extracting structured information from heterogeneous industrial Bills of Materials
- Develop an AI-driven methodology combining deterministic preprocessing with large language model-based extraction to identify, extract, and hierarchically reconstruct the components, attributes, and relationships contained in heterogeneous industrial Bills of Materials, generating a structured, hierarchical, machine-readable representation to support foreground LCI development
- Evaluate the proposed methodology through technical and industrial case studies, and characterize its role and contribution relative to the conventional manual approach to foreground inventory preparation

1.4 Research Questions

The overall research path presented in this thesis is guided by the following questions:

RQ1. How have AI techniques been applied to LCA and LCI generation and what research gaps remain in the automation of foreground LCI development?

RQ2. How can an AI-driven methodology automatically transform heterogeneous industrial BoM into structured representations suitable for foreground LCI development?

RQ3. To what extent can the proposed methodology support foreground LCI development in industrial practice?

1.5 Scope of the Research and contribution

This research focuses on automating the interpretation and structuring of industrial engineering data, specifically Bills of Materials in spreadsheet format, as a preparatory step for foreground Life Cycle Inventory development. The proposed methodology addresses the extraction, organization, and hierarchical reconstruction of engineering information required to produce a structured representation suitable for subsequent LCI use; it does not extend to LCA impact assessment, interpretation of environmental results, uncertainty propagation, or product redesign, all of which lie beyond the scope of this work. Similarly, while the resulting structured representation is designed to support future integration with background databases, the methodology does not itself perform dataset selection or complete environmental modelling. By concentrating on this transformation, this thesis addresses a significant bottleneck limiting the industrial adoption and scalability of Life Cycle Assessment: the manual effort currently required to convert manufacturing-oriented engineering documentation into a form usable for environmental inventory development.

The main scientific and practical contributions of this thesis are summarized as follows.

First, this research provides a comprehensive analysis of the current state of the art on Artificial Intelligence applications in LCA and information extraction for LCI generation. Through a review of recent literature, the study

identifies the capabilities and limitations of existing AI techniques, including ML, NLP, Knowledge Graphs, and LLMs, while highlighting the lack of approaches dedicated to the automated transformation of industrial Bills of Materials into LCI-oriented information.

Second, this thesis proposes an original AI-driven methodology for automatically processing heterogeneous industrial Bills of Materials. Unlike conventional approaches that focus on data extraction alone, the proposed methodology reconstructs product hierarchies, preserves parent–child relationships, and reorganizes manufacturing-oriented information into a standardized, hierarchical, machine-readable representation specifically designed to support foreground LCI development. The methodology combines a deterministic preprocessing stage, which cleans and standardizes heterogeneous BoM formats, with a large language model-based reconstruction stage that interprets structural cues embedded in the BoM, such as level indicators, to infer parent–child relationships and rebuild the product hierarchy. This hybrid design grounds the method in both rule-based reliability and the interpretive flexibility needed to handle inconsistent industrial data formats, providing a consistent basis for subsequent environmental data enrichment while reducing the manual interpretation normally required during inventory preparation.

Third, the proposed methodology is validated through both technical and industrial evaluations. The technical validation shows that component extraction is reliable across BoMs of varying format and terminology, while hierarchy reconstruction is strong in most evaluated cases but degrades in specific, identifiable conditions, very large BoMs, and BoMs lacking an explicit hierarchy indicator. The industrial case study complements this by demonstrating the methodology's applicability to a real manufacturing product, reproducing the physically verified structure of three product modules in full. Together, these evaluations show that the methodology

reliably reduces manual preprocessing effort and generates reusable, LCI-oriented information within a validated and explicitly bounded scope, rather than as a universal guarantee.

Finally, this research establishes a foundation for future AI-assisted LCI automation. Although the proposed methodology does not generate complete Life Cycle Inventories automatically, it creates the structured product representation required for subsequent integration with background LCI databases, semantic matching techniques, and more advanced AI-based environmental assessment workflows.

1.6 Thesis Organization

The remainder of this thesis is organized as follows:

Chapter 2 reviews the state of the art in LCA, with particular emphasis on the LCI generation phase, existing industrial challenges, Life Cycle Assessment databases, and recent Artificial Intelligence techniques relevant to LCA and document extraction for inventory generation. The chapter concludes by identifying the research gap addressed by this work.

Chapter 3 documents the manual foreground LCI generation workflow observed through the Unipr research team's industrial practice, and analyzes the limitations of this practice that motivate the need for automation.

Chapter 4 presents the proposed Artificial Intelligence-driven methodology for automatically transforming heterogeneous industrial Bills of Materials into structured, hierarchical, machine-readable representations that support foreground LCI development, and describes each stage of the developed framework.

Chapter 5 presents the evaluation and validation of the proposed methodology through technical validation and an industrial case study, demonstrating its

applicability to industrial BoMs and its suitability for supporting foreground LCI generation.

Chapter 6 discusses the scientific contributions, industrial implications, and limitations of the proposed research.

2. STATE OF THE ART

2.1 Life Cycle Assessment Methodology

LCA is a systematic, internationally standardized method to evaluate the environmental impact of a product system, process or service throughout all the stages of its life cycle, from material extraction and production, to use and End-of-Life. The methodology follows a series of international standards defined by the ISO, principally within the ISO 14000 standards related to environmental management. The development and alignment of the LCA methodology have been formalized through the ISO 14040 and ISO 14044 standards that define the general principles, framework and methodological requirements of the approach. These standards ensure that the LCA studies are conducted in a scientifically consistent, transparent, and replicable way, allowing the acquisition of comparable results globally. According to (*ISO 14040:2006 - Environmental Management — Life Cycle Assessment — Principles and Framework*, n.d.), the LCA methodology is composed of four phases as shown in Figure 2.1:

The goal and scope definition phase:

This initial phase defines the purpose of the study, the functional unit, and the system boundaries. It forms the basis of the entire evaluation and plays a key role in shaping how the results are understood and applied. the functional unit is a quantified description of the function provided by the product system and acts as the baseline reference for all Life Cycle Assessment calculations, it also provides consistency and comparability between different systems or scenarios. The role of system boundaries is to define which life cycle stages to include in the assessment. Depending on the objective of the project,

system boundaries may cover the full life cycle of the product (cradle-to-grave) or only selected stages, from raw material extraction until production (cradle-to-gate). It can also only include the production stage (gate-to-gate).

The Life Cycle Inventory (LCI):

This second phase is crucial to the LCA framework. In this step, input data related to materials, processes, and energy flows along with output data related to waste and emissions of a product system is collected and quantified.

Life Cycle Impact Assessment (LCIA):

The LCIA phase interprets the possible environmental impact based on the LCI collected data. This phase includes characterization and classification steps. This means assigning data to impact categories such as climate change, ozone depletion, etc., and quantifying impacts in each category using scientific models. Optional elements such as normalization, grouping, and weighting can also be included depending on the goal of the study.

Interpretation:

in this final phase, results from the LCI and LCIA are interpreted and analyzed in accordance with the scope and goal defined in the initial step. The interpretation phase aims to derive conclusions and recommendations from the inventory and impact assessment results. Sensitivity and uncertainty analysis can be added to assess the robustness of the results.



Figure 2.1. The four phases of the Life Cycle Assessment methodology and their interrelation (goal and scope, LCI, LCIA, interpretation)

LCA studies typically distinguish between foreground data, which is data directly related to the product and background data which is data related to external industrial processes present in databases.

Over the last decades, LCA has been applied by industries and policy makers as a decision-making tool. It supports sustainable development of products and environmental labeling by providing a comprehensive assessment of environmental impacts. In product development, it is very used for eco-design, allowing engineers to identify environmental hotspots early in the design stage of the product and optimize material selection, energy consumption, and end-of-life strategies. LCA is also widely applied to compare products environmentally. It allows the evaluation of alternative materials, technologies, and supply chain configurations. In industrial contexts, it supports sustainability strategies by guiding process optimization and resource efficiency improvements. Additionally, LCA plays a key role in regulatory compliance, particularly in the context of environmental reporting, eco-labeling schemes, and European sustainability frameworks. One of its most widespread applications is carbon footprint assessment, where it is used to quantify greenhouse gas emissions associated with products, services, or entire systems in a standardized and comparable way.

Among the four stages of the LCA, the LCI data collection is generally considered the most data-intensive and time-demanding phase (*International Reference Life Cycle Data System (ILCD) Handbook - Publications Office of the EU*, n.d.) The quality of the environmental impact assessment depends a lot on the quality and completeness of the data of the inventory. Consequently, considerable research efforts have focused on enhancing LCI generation and management.

2.2 The Life Cycle Inventory Phase

As introduced in Section 2.1, the LCI phase involves the structured collection and quantification of input and output data associated with a defined product system. This phase forms the essential data backbone for the subsequent LCIA, making the quality, completeness, and accuracy of LCI data critical to the reliability of the overall LCA results. Due to its data-intensive nature, the LCI phase is both complex and resource-demanding. It requires the integration of various data types that capture the environmental exchanges linked to the product across its life cycle.

According to the ISO 14040 and ISO 14044 standards, the objective of the LCI phase is to compile and quantify all relevant elementary flows associated with the system under study. These flows typically include material and energy inputs, emissions to air, water and soil, waste generation, transportation activities, and other exchanges occurring throughout the different stages of the product life cycle. Therefore, the result inventory provides quantified environmental exchanges associated with the product system.

The collection of an LCI generally depends on a process-based representation of the system, in which the product life cycle is decomposed into interconnected unit processes describing activities such as raw material extraction, manufacturing, transportation, product use, and end-of-life treatment. Each process is characterized by quantified input and output flows, allowing the environmental burdens associated with the product system to be evaluated systematically.

Because the environmental impacts calculated during LCIA are directly derived from the inventory flows compiled during the LCI phase, inaccuracies, inconsistencies, or missing information within the inventory

may propagate uncertainty throughout the entire assessment. Consequently, the development of accurate, complete, and representative inventory data is considered one of the most critical aspects of LCA practice. Furthermore, the increasing complexity of industrial products and supply chains has significantly increased the amount and heterogeneity of information required for LCI generation, making this phase one of the most demanding stages of the LCA methodology.

To create a full LCI, both foreground and background data need to be collected. Foreground data is generally categorized into two main types: upstream and downstream data. Upstream data refers to data related to processes and activities that occur before the product's main manufacturing stage. This includes raw material extraction, material processing, and component production. Such data typically comprises material types, weights, and the specific manufacturing processes used to produce the product under study. Additionally, upstream data encompasses the energy consumed during the manufacturing process, usually obtained directly from the production facility. In contrast, downstream data refers to the environmental exchanges that occur after the product has been manufactured. This includes data related to the product's use phase such as energy and water consumption during operation and EoL processes. Downstream data often accounts for the disposal, recycling, or recovery of materials and components, typically expressed in terms of treatment percentages or recovery rates for each material type. Foreground data is provided directly from the manufacturer. Background data is data presenting materials and processes that are collected and presented in databases such as Ecoinvent. Once the foreground data is collected for the LCI, background data is matched with it from LCA databases. The integration of foreground and background data constitutes a fundamental step in LCI modelling. Foreground information collected from the manufacturer must be linked to corresponding background

datasets representing materials, energy carriers, transportation systems, and industrial production processes. This process of data matching is essential to ensure consistency within the inventory model and to accurately represent the environmental impacts associated with the product under study. However, establishing these links is often complex due to differences in terminology, system boundaries, and process descriptions between industrial product information and datasets available in the databases.

The credibility of an LCI depends significantly on the quality and completeness of the data collected. Missing or inaccurate inventory information may significantly affect the validity of the environmental assessment and increase uncertainty in the results of the LCA. In practice, obtaining highly representative inventory data remains challenging, particularly for complex industrial products and global supply chains.

Several factors influence the quality of the data within the LCI phase, including temporal, geographical, and technological representativeness. Background datasets may become outdated due to technological developments in manufacturing systems and industrial processes, while emerging technologies are often insufficiently represented in existing databases. Furthermore, available datasets may not accurately reflect the geographical conditions of the actual production system, potentially limiting the precision of the environmental assessment.

When primary data is unavailable, practitioners frequently rely on assumptions, generic datasets, or proxy values to complete the inventory model. Such approaches facilitate the construction of the LCI, but they may introduce additional uncertainty in the results, which reduces the reliability of the resulting environmental indicators.

As industrial products and manufacturing systems become increasingly complex, the generation of LCI data requires the collection and organization of large volumes of heterogeneous information originating from multiple sources and stakeholders. The diversity of data formats, the fragmentation of industrial information, and the dependence on manual data processing significantly increase the difficulty of LCI generation in industrial practice. These challenges are discussed in the following section.

2.3 LCA Database and Background Data

Life Cycle Assessment results depend strongly on the quality and robustness of LCI data. In particular, background data play a central role in representing upstream and downstream processes that are not directly measured within the system boundary. LCA databases provide structured and quantitative background datasets covering inputs, outputs, resource use, emissions, and waste flows. These databases also support methodological consistency by standardizing data formats, units, and system boundaries, thereby improving comparability across studies and application domains.

Beyond standardization, LCA databases enable the modelling of complex product systems by providing datasets for a wide range of industrial sectors, geographical regions, and technologies. This reduces the need to model background processes from scratch and supports the completion of system boundaries in LCA studies. In addition, many databases provide documentation on data sources and modelling assumptions, which contributes to transparency and reproducibility.

Several established databases are widely used in LCA practice. Among them, Ecoinvent and GaBi are the most frequently applied. The Ecoinvent database is widely used in academic contexts due to its extensive process coverage and detailed documentation, while GaBi is often used in industrial applications

where integration with corporate environmental management systems is required. Both databases provide comprehensive background datasets, although they differ in methodological approaches, system modelling choices, and regional specificity.

Other databases, such as the ELCD, Agribalyse, and the USLCI, provide more region- or sector-specific datasets. ELCD is aligned with European methodological frameworks, Agribalyse focuses on agricultural and food systems with detailed French datasets, and USLCI provides data primarily for North American contexts. These differences highlight the variability of background data sources in terms of scope, coverage, and update frequency.

The selection of a background database depends on the goal and scope of the study, geographical representativeness, and required level of detail. In many LCA applications, the Ecoinvent database is used as a reference background database due to its broad coverage and methodological consistency, as well as its integration with commonly used LCA software tools. For this research, the Ecoinvent database is selected as the primary source of background data to ensure consistency of modelling assumptions and comparability with existing studies.

2.4 Challenges of LCI Generation in Industrial Practices

Although the LCA methodology is standardized, generating the LCI remains a complex operation, particularly in industrial contexts. Challenges arise both from the methodological characteristics of the inventory phase itself and from practical difficulties in data collection and management in real-world environments. This section is structured in three parts: general challenges of LCI generation, operational and practical difficulties in data collection and processing, and challenges specific to industrial environments characterized by fragmented information systems, complex product structures, and

heterogeneous data sources. It closes with a synthesis motivating the AI-driven automation proposed in this thesis.

2.4.1 General Challenges of LCI Generation

LCI generation is inherently data-intensive, describing a product system comprehensively requires collecting and quantifying material inputs, processes, energy flows, and transportation activities across all life stages. Setting up inventory data has long been recognized as one of the most labour- and time-intensive stages of an LCA study, often compounded by a lack of appropriate data for the specific product system (Fang et al., 2024; Finnveden et al., 2009) . This intensity scales with product and process complexity: cradle-to-gate modelling of dual-phase automotive strip steel, for instance, required resolving circular and unidirectional material flows across seven blast furnace plants, five sinter plants, and multiple coke and pellet plants before a single carbon-footprint value could be obtained . As system boundaries expand, the required information grows substantially, and the likelihood of incomplete information, inconsistencies, and data gaps increases correspondingly. This complexity has motivated interest in automating parts of the LCA process; even proponents of applying ML to LCA acknowledge that the sheer number of input parameters and their uncertainties makes broad application difficult, favoring a "divide and conquer" strategy in which ML targets discrete, manageable sub-tasks rather than the system as a whole

Beyond data intensity, LCA result quality depends directly on LCI quality, which is undermined by several sources of uncertainty. Measurement inaccuracies, and data that do not adequately represent the temporal, geographical, or technological characteristics of the studied system, dimensions explicitly built into background databases such as Ecoinvent, which distinguishes datasets by geography, technology, and market versus

production representativeness (Wernet et al., 2016). In a controlled comparison of practitioners modelling the same product system, even after harmonizing system boundaries and process definitions, differences in dataset selection and gap-handling produced measurable divergence in results (Scrucca et al., 2020), demonstrating that inventory-stage choices materially affect outcome quality. Data quality itself is typically assessed against completeness, accuracy, consistency, reliability, and representativeness (Wernet et al., 2016), and gaps or outdated data along these dimensions directly reduce the reliability of LCA results. Because uncertainty cannot be eliminated, managing it remains one of the most significant challenges in developing robust, reliable inventories.

2.4.2 Operational and Practical Difficulties in Data Collection and Processing

Beyond these methodological challenges, LCI development faces substantial practical difficulties. Data required for LCA are rarely centralized: they are spread across production management, maintenance, procurement, and environmental reporting systems that are designed for operational or financial purposes rather than environmental assessment, leaving practitioners to assemble complete datasets manually. Consistent with this, a review of LCA guidance documents and case studies found that only 40% of reviewed studies provide a complete description of their data collection process and a transparent account of their LCI data, indicating that data availability and documentation gaps are widespread (Saavedra-Rubio et al., 2022). Even when companies publish quantitative environmental data, through sustainability or annual reports, this information is often fragmented, inconsistently structured across sources, and provided in non-machine-readable formats such as PDFs combining text, tables, and graphics, further limiting its extraction and reuse (Bejjit et al., 2026).

Once identified, this heterogeneous data (technical reports, spreadsheets, internal databases, supplier documentation, expert interviews) requires considerable manual preprocessing to convert it into life-cycle-oriented flows, since it is generally not standardized for LCA purposes. The effort required scales with the level of detail sought: in an automotive case study, obtaining more precise, granular foreground data from supplier documentation required substantially greater manual workload than relying on coarser, aggregated information (Oliveira et al., 2022). Practitioners frequently revisit sources to clarify ambiguous information or refine estimates, further increasing time and organizational effort, which makes LCI development a labour-intensive task that limits the ability to update inventories frequently in dynamic production environments.

This reliance on manual processing also exposes the inventory to human error, in data entry, unit conversion, and aggregation, particularly when handling large or heterogeneous datasets. While the broader phenomenon of practitioner-dependent variability in LCA outcomes is empirically documented (Scrucca et al., 2020), direct, granular quantification of specific error types such as transcription mistakes is less established in the literature, so this discussion should be read as a plausible, literature-consistent account rather than a separately measured finding. Compounding this, the absence of standardized data collection and recording conventions across organizations and sectors, differing naming conventions, system boundary definitions, and flow reporting practices, complicates dataset integration and introduces additional inconsistencies that require further cleaning, harmonization, and validation before use.

2.4.3 Challenges Specific to Industrial Environments

Industrial environments introduce further, more structural challenges. Product-related information is typically distributed across multiple systems,

ERP, PLM, CAD, MES, supplier databases, each storing information in different data models and formats: ERP systems hold purchasing and material information, PLM platforms focus on product structures and engineering data, and CAD models provide only geometric representations, a division also described directly in the LCA-integration literature (Zhou & Tao, 2021). This heterogeneity is corroborated outside the LCA field: even standardization efforts aimed at closing lifecycle information loops report that existing Product Data Technology standards focus on static product definitions rather than lifecycle needs, and that competing standards (e.g., STEP/PLCS, MIMOSA OSA-EAI, EPCglobal) cover different data types and sectors without a single consistent exchange mechanism (Kiritsis, 2011). Proposed remedies, such as overlaying a domain-agnostic conceptual model of Material, Energy, Information, Equipment, Human, and Environmental flows onto the manufacturing system (Kaushal & Chakrabarti, 2023), improve coverage but still require largely manual mapping rather than eliminating the underlying fragmentation. This interoperability gap is documented specifically for CAD and PLM systems: recent reviews report that CAD-derived environmental results are often simplified due to inherent CAD–LCA interoperability problems, and that although most data theoretically required for LCA exist within PLM systems, coupling LCA functionality with PLM in practice remains complex (GUYON et al., 2024).

Among available information sources, BoMs is one of the most valuable inputs for foreground LCI generation, describing a product's hierarchical composition of assemblies, subassemblies, components, and quantities. However, BoMs are rarely standardized: companies (and even departments within the same company) use different templates, naming conventions, hierarchy representations (indentation, numerical levels, parent-child identifiers), and file formats, often with metadata, empty rows, merged cells, or multiple worksheets requiring manual preprocessing. Furthermore, BoMs

primarily describe product structure rather than environmental information, rarely containing manufacturing processes, transport distances, recycled content, or end-of-life assumptions. Even where a material field exists, it is frequently an internal name or specification code rather than a standardized designation, requiring specialist interpretation before it can be linked to an environmental dataset (Castle et al., 2025), in some sectors, manufacturer-provided BoM data are incomplete or undisclosed altogether, to the point that researchers have reconstructed BoM-level material and mass data through physical disassembly (Babbitt et al., 2020).

Access to this data is further constrained by confidentiality: detailed BoMs, supplier specifications, and production parameters are commercially sensitive, limiting the open datasets available for developing and validating automated LCI methods, unlike many other AI application domains. Much of the remaining expertise needed to construct a complete inventory, selecting representative datasets, interpreting engineering terminology, identifying equivalent materials, is tacit and rarely documented, making the process highly dependent on expert judgement and limiting reproducibility and scalability.

This dependency is most visible in foreground-background matching, where each foreground material, component, or process must be linked to a background dataset (e.g., from Ecoinvent). Industrial descriptions frequently use internal codes or supplier-specific terminology that differ from standardized dataset names, and multiple candidate datasets may appear equally suitable, requiring expert judgement, a manual, judgement-dependent process documented directly in an automotive case study (Oliveira et al., 2022). Underspecified entries (e.g., "plastic housing" without polymer type) compound this ambiguity, motivating recent work on automatically expanding such descriptions before database matching (Castle et al., 2025). This thesis addresses the upstream challenge of extracting and structuring

foreground inventory data from BoMs; extending the framework to automate foreground-background matching is left for future work.

Finally, these challenges scale poorly with product and portfolio complexity. Modern products may contain thousands of components across multiple hierarchical levels, each requiring material identification, process assignment, and database matching, and this effort must be repeated for every product variant as redesigns, supplier changes, and engineering revisions occur, a tension identified more broadly as a barrier to applying LCA at industrial scale, alongside data intensity and system complexity (Kiemel et al., 2022). In sectors with short development cycles (automotive, electronics, consumer goods), the time required to manually generate and maintain inventories often exceeds the pace of product development, limiting the integration of environmental assessment into design decisions.

2.4.4 Synthesis and Motivation for Automation

Collectively, these challenges show that LCI generation remains fragmented across heterogeneous industrial systems, structurally unstandardized in its primary data source (the BoM), constrained by confidentiality and tacit knowledge, and dependent on manual, judgement-based matching to background databases, limitations that intensify with product and portfolio complexity. Existing non-AI methodological responses each address one facet of the problem without resolving it comprehensively: simulation-based LCA embeds the inventory within a dynamic production model to reduce recurring data-collection effort (Liu & Syberfeldt, 2022), process-coefficient-based frameworks formalize relationships among unit processes to ease re-modelling when production paths change (Fang et al., 2024), structured data collection guidance improves transparency and reproducibility of the collection process itself (Saavedra-Rubio et al., 2022), and conceptual, class-based models of the manufacturing system reduce omissions caused by

human judgement (Kaushal & Chakrabarti, 2023). Each nonetheless still depends fundamentally on manual data provision, structuring, or mapping.

These limitations in existing non-AI methodological responses, combined with the structural and informational barriers discussed throughout this section, motivate the growing interest in artificial intelligence, particularly natural language processing and large language models, as a means of automating stages of LCI generation. The extent to which current AI-based approaches address these barriers, and where they remain limited, is examined in the following section.

2.5 Artificial Intelligence in Life Cycle Assessment

The challenges outlined in Section 2.4, particularly the manual, judgement-dependent nature of foreground data structuring and foreground-background matching, have motivated growing interest in artificial intelligence as a means of automating parts of the LCA workflow. This section reviews four major AI paradigms applied within LCA: Machine Learning, Natural Language Processing, the Internet of Things, Knowledge Graphs, and Large Language Models.

▪ Machine Learning Applications in Life Cycle Assessment

ML, a major branch of AI, refers to computational methods that enable computer systems to learn patterns from data and progressively improve their performance without being explicitly programmed for every task. Owing to their ability to analyze large datasets and identify nonlinear relationships, ML techniques have increasingly been adopted within the LCA domain to improve the efficiency, accuracy, and scalability of environmental assessments, and their use has grown sharply over the past several years as more environmental data and computational capacity have become available (Salla et al., 2025).

ML algorithms are generally classified into four categories according to learning strategy, supervised, unsupervised, semi-supervised, and reinforcement learning, with supervised learning dominating LCA research, primarily framed as regression or classification tasks (Romeiko et al., 2024). Among supervised techniques, Artificial Neural Networks (ANNs) are consistently reported as the most widely adopted model across independent reviews, followed by Random Forest and, in agriculture-heavy samples, ANFIS, reflecting the sensitivity of technique choice to sectoral data characteristics rather than a universal algorithmic hierarchy (Romeiko et al., 2024), (Salla et al., 2025). Within the construction sector specifically, ANN and hybrid models have been shown to outperform conventional statistical methods for predicting environmental impacts, emissions, and energy consumption (Ghoroghi et al., 2022), (Salla et al., 2025)).

Beyond impact prediction, ML has also been applied to surrogate modelling, training models on simulation-generated data to approximate computationally expensive environmental calculations for rapid scenario analysis, and to estimating missing energy and material data during LCI compilation (Romeiko et al., 2024). Structurally, however, existing applications concentrate heavily on the Life Cycle Impact Assessment stage, with comparatively few studies targeting inventory compilation directly (Romeiko et al., 2024).

The principal limitation constraining ML in LCA is data scarcity: training datasets are typically small, domain-specific, and drawn from heterogeneous, format-incompatible sources, a constraint substantially more severe than in most other ML application domains (Romeiko et al., 2024), (Salla et al., 2025)). Model evaluation practices also remain limited, with few studies comparing multiple algorithms or reporting model uncertainty systematically (Romeiko et al., 2024).

Overall, ML has significantly advanced environmental impact prediction and surrogate modelling, but existing applications focus predominantly on numerical estimation rather than on automating the extraction and structuring of foreground inventory information from engineering documents, a gap that has motivated growing interest in combining ML with Natural Language Processing and Large Language Models, discussed in the following sections.

- **Natural Language Processing in Life Cycle Assessment**

NLP is a branch of Artificial Intelligence that enables computers to process, interpret, and extract meaningful information from human language. NLP techniques transform unstructured textual information into machine-readable representations through operations such as text cleaning, tokenization, lemmatization, vectorization, and semantic similarity analysis.

These capabilities make NLP particularly suitable for LCA, where a considerable proportion of the required information is contained in engineering documentation, technical reports, product descriptions, environmental declarations, and databases that were not originally developed for environmental modelling.

Within the LCA domain, NLP has increasingly been applied to automate the interpretation of engineering documents, standardize heterogeneous terminology, and facilitate the association of product information with environmental datasets. Unlike conventional Machine Learning methods, which primarily operate on numerical data, NLP techniques enable the processing of textual information that traditionally requires extensive manual interpretation by LCA practitioners.

One representative application was proposed by (Forth et al., 2023), who investigated the use of NLP to automate greenhouse gas (GHG) calculations during the early design stages of buildings. Their methodology integrates BIM with LCA through a process the authors term "semantic model healing":

incomplete or imprecise IFC elements from early-design BIM models are automatically matched to entries in a purpose-built LCA knowledge database (LKdb), which is itself linked to the German LCI database ÖkobaDat. Rather than relying on a single NLP technique, the authors systematically compared three approaches: GermaNet (a lexical-semantic network relying on shortest-path similarity between synsets), spaCy (a pretrained neural embedding model), and BERT (a transformer-based contextual language model) and found that BERT consistently produced the highest and least dispersed cosine-similarity scores, despite not being specifically trained on construction terminology.

The proposed workflow transforms element and material names into vector representations at three levels of granularity: element, material category, and material option and computes similarity using cosine similarity; a decision threshold of 80% is used to determine whether the matched element is trustworthy enough to use directly or whether a generic default element from the knowledge database should be substituted instead (Forth et al., 2023). This explicit thresholding mechanism is an important refinement compared to earlier, less transparent matching pipelines, as it formalizes the trade-off between automation and reliability that is otherwise left implicit in many BIM–LCA integration studies.

The methodology was evaluated using five real-world office building projects. Depending on the weighting scheme applied (by number of elements versus by element surface area, which is more directly relevant to LCA-relevant quantities), correct matching rates ranged from approximately 63% to 89%, with an overall element-weighted accuracy of 78.1% and a combined correctly matched or acceptably defaulted rate of 86.7% (Forth et al., 2023). Crucially, the authors also disaggregated the sources of matching error into five categories: insufficient elements available in the knowledge database,

incorrect or missing DIN 276 classification, insufficient information in the model, and genuine mismatches by the NLP algorithm itself and showed that classification errors, not NLP performance, were the dominant driver of poor matches in several case studies. This finding nuances the interpretation of "matching accuracy" as a metric: it is not purely a measure of NLP performance but is strongly conditioned on the completeness and correctness of upstream BIM classification data, a distinction that is often blurred in the wider literature. Instead of providing a single deterministic result, the framework generates impact ranges that explicitly account for uncertainty during the early design stages validated against a manual eLCA calculation, where the manual single-point result fell within, but toward the lower bound of, the automatically generated range, thereby supporting more transparent sustainability decision-making (Forth et al., 2023).

Despite these promising results, several limitations remain. The methodology is specifically designed for BIM environments and therefore depends on the availability of structured digital building models. Its performance also relies heavily on the quality and completeness of both the engineering models and the underlying environmental knowledge base; the authors note explicitly that incorrect or missing element classification (rather than semantic mismatch) was the most frequent cause of matching failure (Forth et al., 2023). Furthermore, although semantic similarity significantly improves dataset matching, expert verification remains necessary to validate ambiguous cases and ensure the methodological consistency of the resulting Life Cycle Assessment. The authors also acknowledge that using a general-purpose, non-domain-trained BERT model increases computational cost due to high-dimensional (768-dimension) vector representations and suggest that a domain-specific model could reduce this overhead if sufficient training data were available (Forth et al., 2023).

Another important application of Natural Language Processing was presented by (Pham et al., 2024), who combined NLP and Machine Learning techniques to address the heterogeneity of EBM databases. The study recognized that environmental product databases frequently employ different naming conventions, categorization systems, and data structures, making the comparison and integration of information particularly challenging.

To address this problem, the authors first collected more than 300,000 initial records from eight EBM databases using automated web scraping and web crawling techniques, which were subsequently refined to 64,350 usable data points spanning 16 distinct eco-labels (e.g., EPD, HPD, Cradle to Cradle, UL GREENGUARD, Energy Star) (Pham et al., 2024). NLP preprocessing methods including text cleaning, tokenization, stop-word removal, lemmatization, and TF-IDF feature extraction were applied to standardize the collected information before five supervised Machine Learning algorithms (Naive Bayes, Decision Tree, Random Forest, SVM, kNN) were trained and cross-validated for each eco-label individually. Among the evaluated models, Random Forest consistently achieved the highest classification performance across most eco-labels, with F1-scores such as 94.72% (testing) for "PBTs" and 93.25% (testing) for "C2Gate," while SVM outperformed RF for a small number of labels, notably achieving a perfect score for the "UL" indoor-air-quality label and an F1-score of 95.77% for Watersense (Pham et al., 2024). This more granular reporting evaluated per eco-label rather than as a single aggregate accuracy figure is a methodological strength worth highlighting, since it reveals that classifier performance is not uniform across attribute types and is sensitive to class imbalance (e.g., HPD and Energy Star are represented by tens of thousands of samples, while CSR and EPR have only a few hundred), a limitation the authors explicitly acknowledge.

The resulting standardized database considerably improved the consistency and accessibility of environmental product information, reducing the manual effort traditionally required to compare eco-labelled materials. Nevertheless, the proposed framework also presents several limitations. The study focused on a relatively limited set of material categories, making its generalization to other product domains uncertain, and model performance was measurably influenced by class imbalance across eco-labels. In addition, although the authors suggested integrating their methodology with BIM and LCA software platforms, such integration was not implemented or experimentally validated (Pham et al., 2024).

A third and more recent contribution, by (Gachkar et al., 2025), departs from the BIM-centric paradigm shared by the two studies above and instead targets Bills of Quantities (BoQs) directly. The authors argue that BIM-based LCA integration despite its advantages remains constrained in practice by incomplete models, inconsistent Levels of Development, and the frequent unavailability of BIM altogether, particularly for smaller or renovation-type projects. Their proposed framework instead builds a reference database of standardized construction elements and components (using the *Base de Costes de la Construcción de Andalucía*), extracts elements from textual BoQs using pattern-recognition techniques, and matches BoQ elements to reference elements using a Bag-of-Words model and cosine similarity computed on a document-term matrix, restricted within matching construction-work categories to reduce dimensionality (Gachkar et al., 2025). Applied to a real rehabilitation project, the approach achieved an average cosine similarity of 0.54 at the element level (0.98 at the work-category level) and reproduced 30.4% of actual component quantities with zero error, corresponding to an overall price-weighted estimation error of 25.6% while reducing data-gathering time by an estimated 80–90% compared to manual or BIM-assisted workflows (Gachkar et al., 2025).

Notably, the authors deliberately chose the computationally lightweight cosine-similarity/BoW approach over embedding-based methods such as Word2Vec, Doc2Vec, or SBERT, arguing that term-overlap-based similarity is more scalable for large BoQ datasets, even though it does not capture deep semantic meaning as effectively as transformer-based embeddings (Gachkar et al., 2025). This is a relevant methodological contrast to (Forth et al., 2023)'s adoption of BERT: whereas (Forth et al., 2023) prioritized matching accuracy at the cost of computational overhead, (Gachkar et al., 2025). explicitly traded off semantic depth for scalability and transparency, reflecting a broader tension in the reviewed literature between matching precision and computational feasibility that has not yet been resolved by a single unified approach. The authors also candidly acknowledge that their method's effectiveness depends on shared vocabulary and terminological consistency between BoQ and reference database entries a limitation structurally similar to the classification-dependency issue identified by Forth et al., despite the two studies operating on entirely different data structures (BIM elements versus BoQ line items) (Gachkar et al., 2025).

Overall, the reviewed literature demonstrates that Natural Language Processing provides effective tools for extracting, interpreting, and standardizing textual information required during Life Cycle Assessment. NLP significantly reduces manual effort associated with terminology harmonization, semantic matching, and information retrieval from engineering documentation, whether the underlying data source is a BIM model (Forth et al., 2023), an eco-label database (Pham et al., 2024), or a Bill of Quantities (Gachkar et al., 2025). However, current applications remain largely focused on the construction sector and primarily address isolated tasks semantic matching of elements or materials to reference datasets, or classification of environmental attributes rather than the full inventory-generation problem.

More importantly, and consistently across all three studies reviewed here, existing NLP approaches generally process textual descriptions independently and do not reconstruct the hierarchical engineering structures required for foreground Life Cycle Inventory generation from industrial Bills of Materials. Even (Gachkar et al., 2025) 's BoQ-based framework arguably the closest of the three to the goal of automated inventory construction, since it directly produces a "decomposed price table" analogous to a bill of materials still depends on matching individual BoQ line items independently to a flat reference database, rather than reconstructing the compositional and hierarchical dependencies between components that a genuine Bill of Materials requires; the authors' own estimation error of 25.6% reflects, in part, this structural limitation rather than semantic mismatch alone (Gachkar et al., 2025). Consequently, while NLP substantially improves the interpretation of heterogeneous engineering information, it does not fully address the broader challenge of automatically extracting and structuring complete foreground inventory models. This limitation has motivated the increasing adoption of Large Language Models, which extend NLP capabilities by combining language understanding with contextual reasoning and structured information extraction.

▪ **Internet of Things in Life Cycle Assessment**

The Internet of Things (IoT) refers to interconnected networks of physical devices equipped with sensors and communication technologies capable of collecting, transmitting, and exchanging data in real time. Within Life Cycle Assessment (LCA), IoT technologies have attracted increasing attention because they enable continuous monitoring of industrial processes and provide direct measurements of resource consumption, energy use, emissions, and other environmental parameters. By reducing reliance on manually collected or estimated information, IoT offers the potential to improve the

quality, representativeness, and timeliness of Life Cycle Inventory (LCI) data.

One of the principal advantages of IoT in LCA is its ability to generate high-resolution operational data throughout the product life cycle. Rather than relying exclusively on historical records or average database values, IoT sensors can continuously monitor manufacturing activities, capturing both direct measurements and proxy indicators such as temperature, machine operating conditions, or energy consumption. These data streams provide a more detailed representation of industrial processes and enable environmental assessments that better reflect actual operating conditions.

(Mba Wright et al., 2024) highlighted the potential of integrating IoT technologies with Machine Learning and blockchain to create traceable, dynamic, and real-time Life Cycle Assessment frameworks. In the proposed approach, IoT sensors continuously acquire operational data, while Machine Learning techniques analyze these measurements to support environmental modelling and decision-making. Blockchain technologies further enhance the framework by ensuring the traceability and integrity of collected data throughout the product life cycle.

Beyond supporting environmental impact assessment, IoT-generated information can also contribute to Life Cycle Inventory development. Continuous measurements collected during manufacturing can be used to estimate energy consumption, resource use, and emissions, while the resulting datasets may subsequently serve as training data for Artificial Intelligence models aimed at improving inventory generation and environmental prediction.

Despite these advantages, several limitations restrict the practical application of IoT within Life Cycle Assessment. First, IoT systems primarily capture data generated within manufacturing facilities and therefore provide only

partial coverage of the complete product life cycle. Information relating to suppliers, upstream material production, and external processes often remains unavailable, limiting the completeness of the resulting inventory. Second, the deployment of sensor infrastructures requires significant technological investment and organizational integration, making implementation challenging for many industrial environments. Finally, the collected datasets are generally proprietary and company-specific, reducing their availability for broader research applications or for training generalized Artificial Intelligence models.

Overall, IoT technologies offer significant opportunities to improve the quality and temporal resolution of environmental data by enabling continuous monitoring of industrial processes.

However, because their applicability remains largely restricted to operational environments under direct organizational control, they cannot independently address the broader challenge of generating complete foreground Life Cycle Inventories across heterogeneous industrial supply chains. Consequently, IoT is best viewed as a complementary technology capable of enriching inventory data rather than replacing the need for automated extraction and interpretation of engineering documentation.

▪ **Knowledge Graphs in Life Cycle Assessment**

KGs are semantic knowledge representation frameworks that organize information as interconnected entities and relationships, enabling both humans and computer systems to interpret complex data structures. Traditional LCI databases function primarily as repositories of numerical information with limited semantic representation of the relationships between products, processes, and calculation methods; KGs address this by explicitly modelling those relationships in machine-readable form, supporting information integration and automated reasoning (Saad et al., 2023).

Recent contributions illustrate a layered progression in how KGs have been applied to LCA. At the infrastructural level, (Saad et al., 2023) proposed a labeled property graph model for representing Ecoinvent data in a graph database, demonstrating substantially more efficient storage and querying than the relational backends used by conventional LCA software, though without automating any LCA task or performing reasoning over the data. Building on this kind of representation, (Fang et al., 2024) used a KG together with a domain-specific thesaurus to automate background-dataset recommendation, substantially improving both the precision and speed of the practitioner-facing search-and-matching step relative to conventional database search. Addressing the LCI stage more directly at the design phase, (Greif et al., 2025) developed a KG-based framework spanning engineering, production, usage, and disposal phases, incorporating large language models to generate synthetic estimates for missing environmental data where empirical LCI data are typically unavailable. Targeting the LCIA stage instead, (Diamantini et al., 2025) proposed a semantic framework representing the mathematical structure of environmental impact indicators themselves, enabling automated reasoning to derive alternative computational expressions when some input data are unavailable — a qualitatively stronger form of quality assurance than the structural checks used in the other three frameworks.

Together, these contributions show KGs supporting distinct parts of the LCA workflow, data infrastructure, dataset recommendation, design-phase data enrichment, and indicator-level reasoning, without yet being combined into a single framework addressing all of these simultaneously. Despite this potential, KG-based approaches share important limitations: constructing and maintaining comprehensive ontologies requires substantial domain expertise and manual effort, particularly in data-sparse conditions where automated extraction proves insufficient (Greif et al., 2025), LLM-based data

enrichment introduces its own estimation uncertainty (Greif et al., 2025), recommendation-based approaches remain bounded by the completeness of the underlying database (Peng et al., 2024), and validation across these studies has generally been limited to single case studies, with generalization to other sectors identified as future work ((Greif et al., 2025),(Peng et al., 2024), (Diamantini et al., 2025)).

Existing KG applications in LCA focus primarily on knowledge integration, retrieval efficiency, and environmental reasoning, rather than on the automated extraction and hierarchical structuring of engineering information from industrial documents such as BoMs.

Large Language Models in Life Cycle Assessment

With the development of Large Language Models (LLMs), they became a trusted tool to merge with other AI techniques to give better, faster and cleaner data. Many researchers have employed LLMs to find data, clean it, or explain it to the AI in order to optimize results for instance, by combining retrieval-augmented generation with Text2SQL-based database querying and automated report generation to build integrated, LCA-domain-specific LLM pipelines (Zhang et al., 2025). LLMs are neural network architectures typically utilizing variants of recurrent neural networks such as the transformer, which are trained on large bodies of textual data using techniques such as pre-training and fine-tuning (Damla Turhan, n.d.).

Preuss et al. investigated the integration of LLMs in the LCA workflow. They mention that LCA is a powerful tool for assessing environmental impact, but it is slow, resource-intensive and prone to error. To address these limitations, they propose the use of LLMs in different stages of the LCA methodology. In the goal and scope definition phase, they suggest that LLMs can assist in framing the goal, choosing the system boundaries, and suggesting functional

units. They argue that an LCA-trained LLM could eventually provide expert-level recommendations, compare contexts and suggesting suitable approaches without the practitioner having to search extensively. Until such models are developed, existing LLMs can still support novices by applying rules of thumb through techniques like few-shot learning, thus making LCA more accessible for non-experts (Preuss et al., 2024).

A paper written by Turhan explores the integration of LLMs such as ChatGPT in LCA to support the evaluation of unconventional biobased construction materials. The research applied the methodology to a case study on a bacterial cellulose composite used as an innovative construction material. The process begins with the definition of a functional unit. The assessment follows a cradle-to-grave perspective, considering raw material cultivation, production processes, transportation, use phase, and end-of-life scenarios. Data collection for the LCI is supported not only by traditional sources but also by queries to an LLM, which is employed to synthesize information on material properties, energy use, and potential impacts, thereby filling knowledge gaps where conventional databases lack coverage. The CML 2001 method is applied to quantify environmental effects such as greenhouse gas emissions, resource depletion, and toxicity. To operationalize this approach, the study further introduces a browser-based prototype (bioLCA) that integrates the LLM via an API, allowing practitioners to interactively explore materials, receive structured impact results, and obtain textual interpretations and optimization suggestions. This methodology demonstrates how LLMs can augment LCA workflows by accelerating data acquisition, supporting interpretation, and making sustainability assessment more accessible within design practice (Damla Turhan, n.d.).

While (Preuss et al., 2024) and (Damla Turhan, n.d.) address LLM integration across the LCA workflow broadly, Tu et al. frame the problem more narrowly

around Life Cycle Inventory (LCI) modeling specifically, identifying two grand challenges that limit LCA accuracy: (1) the creation of missing foreground flow data, and (2) the matching of foreground flows to appropriate background database entries (the "flow-provider" pairing problem). They argue that traditional mechanistic approaches (e.g., process simulation in AspenPlus) and existing similarity-based machine learning methods are both limited in scalability and generalizability the former due to a high domain-knowledge barrier and case-specific validity, the latter due to instability across varying training data compositions and missing-data extents. LLMs are positioned as a potentially scalable alternative precisely because their pretraining already encodes broad domain knowledge, and because retrieval-augmented generation and multimodal analysis allow them to draw on diverse, continuously updatable data sources rather than a fixed, hand-curated training set (Damla Turhan, n.d.).

(Castle et al., 2025) provides a concrete, empirically evaluated implementation of this second challenge background data matching applied directly to the task this thesis addresses: linking components from industrial Bills of Materials (BoMs) to entries in an LCA database (Ecoinvent). Their pipeline uses an LLM (a locally hosted Llama 3.1, 8B) to generate a textual description of the likely manufacturing process for each BOM component from its name, supplier, and material fields (optionally enriched with component datasheet text), and then ranks candidate Ecoinvent entries by the semantic similarity between this generated description and the database's process descriptions. Evaluated against a small set of 21 labeled BOM components, their LLM-based approach matched or slightly exceeded non-expert human performance (Hits@1 of 0.24 versus 0.19 for humans) and clearly outperformed a semantic-similarity-only baseline that skipped the LLM step entirely (Hits@1 of 0.00). This result is important for two reasons: first, it demonstrates that LLM-generated intermediate descriptions add

genuine value over direct embedding-based matching of raw BOM text; second, the authors note that their evaluation was constrained to a very small dataset because of data privacy and trade-secrecy limitations common to industrial BOM data a scalability gap that motivates further investigation into automated, LLM-assisted foreground LCI generation from BOMs (Castle et al., 2025). Linking BOM components to Ecoinvent entries is itself an instance of a much broader and more mature research problem: ontology matching (OM), which aims to identify semantic correspondences between concepts drawn from heterogeneous knowledge sources. Babaei Giglou et al. show that this broader problem can be addressed effectively with LLMs by combining retrieval-augmented generation (RAG) with LLM-based classification: candidate target concepts are first narrowed down using an embedding-based retriever, and only the resulting candidate pairs are verified by prompting an LLM to judge equivalence. Evaluated across 20 datasets from the Ontology Alignment Evaluation Initiative (OAEI), spanning domains from anatomy to materials science, this retrieve-then-prompt framework matched or surpassed traditional, non-LLM OM systems in several complex matching tracks, lending independent, cross-domain support to the premise that LLM-assisted matching can outperform pure similarity-based methods the same premise underlying Castle et al.'s BOM-Ecoinvent pipeline (Giglou et al., 2024).

However, querying an LLM for every retrieved candidate pair scales poorly: Taboada et al. note that this naive retrieve-then-prompt strategy carries a high computational overhead, since each of the k retrieved candidates per source entity requires a separate, costly LLM call. Their proposed system, MILA, addresses this directly by introducing an intermediate retrieve-identify-prompt step: candidate pairs that are each other's top-ranked match in both directions are accepted automatically as high-confidence correspondences without any LLM query, and the LLM is reserved only for the remaining, genuinely ambiguous cases, searched via a prioritized depth-first strategy that

stops as soon as a valid match is confirmed. Tested on OAEI 2024 biomedical, anatomy, and biodiversity tasks, MILA achieved the highest F-Measure in five of seven unsupervised matching tasks, outperforming state-of-the-art OM systems by up to 17%, while cutting runtime dramatically for example, completing the large SNOMED-FMA matching task in roughly 45 minutes, compared to nearly 47 hours for the equivalent naive retrieve-then-prompt pipeline using the same LLM [Taboada et al., 2025]. This efficiency gain is directly relevant to the scalability limitation Castle et al. themselves acknowledge: their evaluation was restricted to 21 labeled BOM components because of data privacy and trade-secrecy constraints on industrial data, leaving open how an LLM-based matching pipeline would perform, in terms of both accuracy and computational cost, on the far larger and more heterogeneous BOMs found in industrial practice. MILA's query-minimization strategy suggests one concrete direction for addressing that gap.

Traditional machine learning (ML) and deep learning (DL) techniques generally require training on labeled or unlabeled datasets before they can be effectively deployed. In contrast, rule-based and knowledge-based approaches do not inherently require model training, as they rely on predefined rules or structured domain knowledge. Consequently, one of the main limitations of applying conventional ML and DL techniques to Life Cycle Assessment (LCA), and particularly to Life Cycle Inventory (LCI) generation, is the limited availability of high-quality domain-specific datasets required for model training.

Large Language Models (LLMs), on the other hand, undergo extensive pretraining on large-scale datasets before deployment. Once pretrained, they can often be applied to downstream tasks through prompt engineering or retrieval-augmented generation (RAG), without requiring task-specific retraining or fine-tuning. This characteristic makes LLMs particularly

attractive for LCI-related applications, where sufficiently large and well-annotated training datasets are often unavailable. As shown by Castle et al., however, existing work in this space remains limited by small, non-public evaluation datasets, leaving open the question of how well LLM-based approaches generalize across larger and more diverse BOMs [Castle et al., 2025]. For this reason, this thesis focuses on the application of LLMs to support the automation of foreground LCI generation from industrial Bills of Materials (BoMs).

2.6 AI Techniques for Structured Information Extraction for LCI Generation

Section 2.5 reviewed the application of Machine Learning, Natural Language Processing, Knowledge Graphs, and Large Language Models across the LCA framework more broadly and closed by noting that LLM applications in this space have concentrated on foreground-background matching and on estimating missing numerical flows, rather than on the automated structuring of foreground information itself. This section examines that narrower question directly: what does the literature currently offer for AI-based extraction of *structured*, and specifically *hierarchical*, information from engineering documents such as industrial BoMs and where does it fall short of the problem this thesis addresses.

2.6.1 Schema-guided structured extraction from documents

A distinct line of work within the broader LLM literature addresses the extraction of structured records from documents through an explicit schema, rather than open-ended text generation. LMDX (Perot et al., 2024) is the most directly relevant example: it reframes information extraction from visually rich documents, invoices, receipts, forms, as a text-in/text-out task guided by a schema embedded in the prompt, supporting singular, repeated, and

hierarchical entity types without requiring architectural changes to the underlying LLM. Because its target documents are OCR'd and spatially complex, LMDX additionally encodes layout through coordinate tokens and verifies extracted values against their source location to reduce hallucination.

Bai et al. (2024) address a closely related but distinct task: transforming *tabular* data into structured records that follow a human-authored schema, evaluated across a benchmark of heterogeneous tables spanning multiple domains. Unlike LMDX, the input here is already tabular rather than visually embedded in a page layout, which removes the need for spatial grounding but leaves open the question of how implicit structural relationships within the table such as row-to-row hierarchy should be recovered. (Dagdelen et al., 2024) provides a further, high-impact precedent for schema-constrained extraction: fine-tuned LLMs (GPT-3, Llama-2) producing structured JSON records from unstructured scientific text in materials chemistry, explicitly designed as a flexible route to building large, structured databases without task-specific architectures.

Together, these three contributions establish that schema-guided, LLM-based structured extraction is an active area with demonstrated effectiveness across visually rich documents (LMDX), heterogeneous tables (Bai et al., 2024), and unstructured scientific text (Dagdelen et al., 2024).

2.6.2 The structured/hierarchy-aware gap

None of the approaches reviewed above directly addresses the specific combination of characteristics presented by industrial BoMs. LMDX operates on documents where hierarchy, when present, must be inferred from spatial and semantic proximity, since no equivalent to an explicit structural field is available; it is not designed to exploit a pre-existing indicator such as a BoM's Level attribute. Bai et al.'s schema-driven extraction targets flat structured

records and does not address the reconstruction of parent-child relationships across rows. More broadly, current AI-based approaches to document and table extraction generally focus on recovering isolated entities rather than reconstructing the compositional, hierarchical relationships between them a limitation directly relevant to LCI generation, where preserving the organization of assemblies and components is essential, since product structure determines how foreground data are aggregated and linked to environmental modelling.

A further, related gap concerns the integration of deterministic data-processing techniques with LLM-based reasoning. Rule-based methods provide robustness for tasks such as table detection, metadata extraction, and data cleaning, while LLMs offer flexibility for interpreting heterogeneous industrial data and generating structured output; the literature reviewed here contains limited work that combines these complementary capabilities into a unified workflow specifically designed for industrial BoM processing.

2.6.3 AI methods applied to LCI/BoM-adjacent problems

Two contributions apply AI methods directly within the LCI domain, though neither addresses hierarchical structuring from raw BoMs.

(Bejjit et al., 2026) develop a semi-automated framework for extracting LCI-relevant quantitative data from company sustainability reports, combining deterministic document preprocessing with a constrained, multimodal (vision-and-text) LLM extraction step operating under zero-temperature, schema-enforced prompting. Applied to copper and nickel production reports, the framework extracted over 100,000 metrics with reported accuracy near 99% on correctly extracted numeric values, while requiring human validation at multiple stages. Methodologically, this is the clearest existing precedent for the hybrid deterministic-preprocessing-plus-constrained-LLM

architecture. However, its extraction target is deliberately flat and atomic each metric independent, with no relationships preserved between them which is the functional opposite of the hierarchy-preserving representation required for foreground product structure.

(Bertrand et al., 2025a) address the downstream problem of automating foreground–background data reconciliation for Electrical and Electronic Equipment products, using semantic similarity matching against a curated library of materials and processes to link BoM entries with background environmental data, explicitly aiming to reduce the several-months timeline typically required for this reconciliation. Their review of prior automation attempts including data-mining-based node translation (Marwah et al., 2011), semantic-network-based linking (Kuczenski et al., 2021) and knowledge-graph-based matching (Peng et al., 2024) confirms that automation efforts in this space have concentrated specifically on foreground–background matching. None of these approaches, including Bertrand et al.'s own method, addresses the prior step of extracting and structuring the foreground BoM itself; the BoM is uniformly treated as an already-available, already-structured input to the matching process, not as heterogeneous raw data requiring interpretation.

Read together, Bejjit et al. and Bertrand et al. demonstrate that AI-based automation within the LCI domain has so far concentrated on two problems adjacent to, but distinct from, the one this thesis addresses: extracting flat environmental metrics from unstructured reports, and matching already-structured foreground data to background databases. Neither engages with the transformation of heterogeneous, non-standardized industrial BoMs into a structured, hierarchical foreground representation in the first place.

2.7 Synthesis and Research Gap

The literature reviewed in this chapter demonstrates substantial progress in applying artificial intelligence to LCA. Machine learning techniques have been successfully applied to impact prediction, uncertainty analysis, and missing-data estimation (2.5), schema-guided LLM extraction has proven effective across visually rich documents, heterogeneous tables, and unstructured scientific text (2.6.1); and AI methods have begun to be applied directly within the LCI domain, for flat environmental-metric extraction and for foreground–background matching (2.6.3).

Despite this progress, the automation of foreground LCI generation from industrial BoMs specifically remains insufficiently addressed. Existing structured-extraction literature does not exploit, or need to exploit, a pre-existing hierarchical signal comparable to a BoM's Level attribute, since the document types it targets (invoices, general tables, scientific text) do not contain one. Existing LCI-domain AI applications either produce deliberately flat, atomic output (Bejjit et al., 2026) or assume a structured foreground representation already exists as their starting point (Castle et al., 2025) , (Bertrand et al., 2025a). Consequently, no reviewed approach addresses the specific combination this thesis requires: reconstructing hierarchical product structure from structured, non-standardized industrial BoMs, using the Level attribute as an explicit reconstruction signal that the document-understanding and flat-extraction literatures have no equivalent reason to exploit.

Based on these observations, this thesis identifies the need for a methodology specifically tailored to structured industrial BoMs, addressing four requirements: (i) automatic detection and extraction of BoM tables from heterogeneous Excel files, (ii) identification and preservation of contextual metadata, (iii) reconstruction of hierarchical product structures from

engineering attributes, and (iv) generation of a standardized, machine-readable representation suitable for foreground LCI development. This literature-based gap is compounded, as Chapter 3 documents through the industrial experience conducted by the Unipr research team industrial experience, by practical realities of BoM heterogeneity, incomplete documentation, and the absence of a pre-existing structured intermediary reinforcing, from an independent evidentiary basis, the same conclusion reached here. Addressing this combined gap constitutes the primary contribution of the research presented in the following chapters.

3. WORKFLOW of a MANUAL LCI GENERATION in INTERNAL PRACTICE

The account presented in this chapter is grounded in industrial LCA practice conducted by the UNIPR research team within the household appliance sector, across several projects covering ovens, freestanding cookers, and microwave products, each typically spanning one to two months. This industrial context is directly related to the case study presented in Chapter 5. Because the observations below come from direct professional practice rather than a documented case study, they are distinguished explicitly from the literature-based discussion that surrounds them: the workflow, sequence, and general characteristics described below are accurate, but specific figures not drawn from project files are noted as such where they occur. The chapter returns at the end of the section to what this sector-specific evidence base can and cannot support.

3.1.1 The manual workflow

Across these projects, the manual workflow followed a consistent sequence. The process began with receipt of the BoM for the product under study, provided directly by the manufacturing company. The BoM was analyzed to identify its structure and, specifically, the parent–child relationships it encoded, which were used to regroup individual components into their corresponding assemblies and subassemblies.

The disassembly was not arbitrary. It began by identifying the real-product assemblies and subassemblies and their correspondence to those defined in the BoM, proceeding from external assemblies, such as the product body, progressively toward internal components. For each component, materials and manufacturing processes were assigned using the engineering expertise of the Unipr research team. Where materials were difficult to identify,

ambiguous cases were resolved through reasoned assumptions, which were subsequently discussed and confirmed with the company rather than left undocumented. A documented instance of this practice concerns electronic boards and components, for which no supplier data were accessible in one project, generic Ecoinvent datasets were adopted as an explicit, reported assumption rather than left unresolved (Brahem et al., 2024). In each project, this concerned a comparatively small subset of components, typically on the order of a handful up to about ten or more, varying by project, rather than the component list as a whole. This step depends substantially on expert judgement, which can introduce variability into the results. Component mass, by contrast, was determined directly through weighing in every project, rather than through indirect estimation. The resulting information was consolidated into a structured Excel file, which then served as the basis for matching foreground components to background datasets in SimaPro.

In practice, BoM analysis and physical disassembly were not sequential, one-off steps but proceeded iteratively over several weeks within the one- to two-month project timeline, with the two activities repeatedly cross-checked against each other. This iterative reconciliation typically required dedicated meetings with the manufacturer to confirm that the hierarchy and materials understood from the BoM genuinely corresponded to the real product, that is, to establish where each assembly and component actually sat within the physical product, and to verify material assignments that could not be resolved from the BoM or visual inspection alone. For example, a comparison across three structurally similar range models found that one variant lacked an insulation material present in the other two, a difference that required disassembly-based verification to confirm, since it was not resolvable from the BoM alone (Brahem & Favi, 2025).

This sequence, document analysis, hierarchy interpretation, physical disassembly, component-level verification, mass and material determination,

and inventory compilation, is broadly consistent with the general workflow reported in the literature. (Babbitt et al., 2020) documented an equivalent process for consumer electronics, generating bill-of-materials data for 95 products through manual disassembly and physical characterization, and noting that manufacturers rarely disclose full product composition, making disassembly the practical means of obtaining it. The projects reviewed here match this account: BoM and any complementary documentation proved insufficient on their own, and physical disassembly was consistently required to verify and complete the product information.

Overall, the manual inventory generation process combined engineering-document analysis, product dismantling, material and process identification, mass determination, and inventory compilation. This matches the general structure reported in the literature while adding a directly observed, sector-specific account of how each stage was actually carried out. It is, however, only one of several workflows followed by LCA practitioners more broadly.

3.1.2 Bills of Materials and CAD files as data sources

CAD files and BoMs were provided directly by the manufacturing company. CAD files supplied geometric and dimensional data. BoMs were delivered in Excel format, the internal team or system that generated them was not disclosed, so the BoM was received as a finished artefact rather than as the output of a process the team had visibility into. Consistent with the heterogeneity discussed in Section 2.4.7, BoM structure differed across projects, even across products from the same sector and, in some cases, the same company, with column layouts, naming conventions, and overall organization varying from file to file. This variation extended even to structurally similar products from the same manufacturer: three freestanding-kitchen models differing only in hob technology (electric, induction, and gas)

required distinct BoM interpretation because of technology-specific sub-assemblies, such as the motherboard and cooling components present only in the induction model (Brahem & Favi, 2025). One structural element remained consistent throughout: product hierarchy was represented by a numeric Level column indicating the nesting depth of each component. The BoMs reviewed were of moderate to substantial size, typically around 150 to 200 line entries and occasionally more, reflecting the structural complexity typical of household appliance products.

The BoM guided the dismantling process directly and let the team identify the components expected within the product and ensure that disassembly followed the product's original architecture. In practice, this typically meant grouping the BoM into a small number of top-level assemblies, an oven assembly, a hob assembly, and packaging, each decomposed into sub-assemblies such as the body structure, cavity, and door for the oven, or the heating elements, front panel, and top panel for the hob, consistent with the BoM-derived structure reported for the freestanding cooker projects reviewed here ((Brahem et al., 2024), (Brahem & Favi, 2025)). By maintaining consistency between the engineered product structure and the inventory representation, the BoM supported the completeness and traceability of the foreground system definition, a role consistent with the literature, where BoM-derived product-information flows have similarly been reported as the starting point for systematic dataset selection in other industrial sectors (Sturm et al., 2025).

CAD files played a secondary role: they were sometimes used to cross-check product structure during dismantling, occasionally provided material or weight information, and in several instances were not usable for inventory purposes at all. In practice, CAD files alone were not sufficient to conduct data collection or support manual disassembly, the BoM was of greater use across every project in this review.

3.1.3 Limitations of BoM data for LCI purposes

As discussed in the literature documents several limitations of BoM data for LCI purposes, and the same limitations were encountered directly across the projects reviewed here. Table 3.1 summarizes them.

Table 3.1 BoM limitations encountered during manual LCI generation, and how each was addressed in practice

Limitation	Manifestation in the reviewed projects	How the gap was closed
Incomplete manufacturer disclosure of composition	Manufacturers did not always disclose the full material composition of a product within the BoM.	Closed through physical disassembly and direct mass determination by weighing , rather than through the document itself.
Ambiguous or internally coded material fields	Material fields were, in some cases, encoded or otherwise ambiguous, with no naming convention consistent across projects.	Resolved through the engineering expertise of the practitioner and the project team; no generic decoding rule could be applied consistently across projects.
Absence of manufacturing-process information	BoMs generally did not include information on the manufacturing processes associated with each component.	Resolved through the engineering expertise of the practitioner and the project team.
Variability in naming of BoM components	Components are defined by variable names, codes and characteristics	Resolved through the engineering expertise of the practitioner and the project team.
Inconsistent presentation of	Hierarchy is defined but its presentation differs from one	Practitioners read the BoM and manually extract the

hierarchy across BoMs	BoM to another, it is not always visually clear	assemblies and subassemblies groups
--------------------------	--	--

Available BoM and CAD information was not always sufficient to complete the inventory, therefore additional meetings with the company were regularly required, and manual disassembly remained necessary even when digital documentation was available. In fact, across the projects reviewed, most components required some form of disassembly-based verification before their composition could be considered confirmed, rather than being resolved from the BoM alone. As Table 3.1 shows, gaps in composition information was closed through physical dismantling and mass determination rather than through the document itself, while ambiguous material fields depended on the same engineering expertise described in Section 3.1.1, since no generic decoding rule could be applied consistently across projects.

Taken together, and consistent with both the literature and the observations reported here, BoMs constituted the structural backbone of the foreground inventory across the projects reviewed: they provided the reference structure around which physical verification, material identification, and process assignment were organized, rather than functioning as a self-sufficient source of all information needed for LCI.

The methodology proposed in this thesis accordingly focuses primarily on the BoM as the main structured input for foreground inventory generation, consistent with its role as the most directly accessible and structurally reliable source observed in practice. The approach is designed to remain extensible: future work could integrate additional sources, particularly CAD models, whose contribution and inconsistent usability were directly observed above.

3.1.4 Foreground–background matching

In the projects reviewed, foreground-to-background matching was performed in SimaPro, using Ecoinvent as the background data source. It is worth being explicit that this describes prior professional practice, undertaken independently of the methodology proposed in this thesis; Ecoinvent/background matching is not implemented in the automated methodology developed here and is treated as future work. For each foreground component or material identified during disassembly and characterization, candidate background datasets were compared and selected based on engineering judgement, considering material specificity, process representativeness, and consistency with the component's identified characteristics.

This matching step was consistently one that required expert judgement and could not be straightforwardly delegated to automation as currently available. This is consistent with the sources of variability discussed in the literature review, and specifically with the practitioner-dependent variability demonstrated empirically by (Scrucca et al., 2020) at the dataset-selection stage. In practice, this variability took a concrete form: rather than a single lookup, dataset selection typically involved comparing several candidate datasets against the same foreground description, with the final choice resting on the individual practitioner's judgement of which candidate came closest, the same pattern reported by de (Oliveira et al., 2022), now corroborated by direct professional observation in a different industrial sector. This is illustrated directly in the projects reviewed: for electronic components such as printed circuit boards, where no supplier-specific data were available, a generic Ecoinvent dataset was selected as the closest available match, a choice explicitly reported as carrying comparatively high uncertainty ((Brahem et al., 2024),(Brahem & Favi, 2025)).

Inconsistencies in naming conventions, variation in material descriptions, and the lack of direct correspondence between engineering components and

database entries introduced further inconsistency into the matching process. Foreground–background integration is a necessary step in LCI modelling, linking product-specific information to environmental datasets; although it is grounded in established LCA practice and expert knowledge, both the literature and the observations reported here confirm that it remains a manual, judgement-dependent process that current automated approaches have not been shown to replace reliably. This conclusion directly informs the scope of the methodology proposed in this thesis, which addresses BoM interpretation and structuring rather than foreground–background matching.

3.1.5 Synthesis and motivation for the proposed methodology

The manual LCI generation process documented here, through both the literature and direct industrial experience, involves repetitive data-management tasks, partially automatable estimation steps, and expert-dependent decision-making, a distinction broadly supported by existing reviews of automation in LCI analysis (Köck et al., 2023) and reinforced by professional observation across multiple household appliance projects. Within this evidence, the processing of engineering documents and the extraction of information while respecting product hierarchy consistently emerges as the activity offering the most concrete and immediate opportunity for automation.

This conclusion reflects a recurring realization across the successive industrial projects described in this chapter: automating foreground LCI generation depends, as a necessary first step, on automating the interpretation, extraction, and structuring of BoM content. Without a reliable, structured representation of the product hierarchy, none of the subsequent steps, material identification, process assignment, or database matching, can be meaningfully supported by automation.

The scale of the manual effort this represents, while not tracked with precision at the time and reported here from recollection rather than from project records, gives an approximate sense of the burden involved. Each project ran to roughly one to two months, of which BoM analysis and physical disassembly were not discrete phases but proceeded iteratively over several weeks, repeatedly cross-checked against each other through meetings with the manufacturer. Within a BoM of some 150 to 200 line entries, most components required disassembly-based verification rather than being resolved from documentation alone, and a smaller subset, typically on the order of a handful up to about ten per project, required additional reasoned judgement to resolve ambiguous material identification. It is this repeated, iterative reconciliation between document and product, rather than any single step in isolation, that the proposed methodology targets.

This account is based on a limited evidence base: the UNIPR research team's experience across a small number of projects in a single sector (household appliances). It should be read as an illustrative, first-hand case that motivates the methodology, rather than as a representative or generalizable survey of manual LCI practice across the industry. Where possible, this is why the discussion above has been anchored explicitly to the broader literature ((Babbitt et al., 2020), (Scrucca et al., 2020), (Oliveira et al., 2022), (Köck et al., 2023)) rather than presented as evidence on its own.

Building on these observations, chapter introduces the proposed methodology, which focuses on the automated extraction and structuring of foreground LCI data from engineering documents, with particular emphasis on BoM processing. The objective is to support and partially automate the transformation of engineering data into structured inventory representations, reducing manual effort in the early stages of LCI generation, directly addressing the bottleneck identified through the industrial experience reviewed above.

4. PROPOSED AI-DRIVEN METHODOLOGY FOR FOREGROUND LCI-ORIENTED REPRESENTATION

4.1 Overview of the proposed methodology

The literature reviewed in Chapter 2 established that existing AI-based structured-extraction approaches whether applied to visually rich documents, heterogeneous tables, or unstructured scientific and environmental text, do not address the reconstruction of hierarchical product structure from semi-structured, non-standardized industrial BoMs, since none exploits a pre-existing structural signal comparable to a BoM's Level attribute.

The proposed methodology employs a two-stage computational architecture: Stage I, deterministic preprocessing and prompt preparation, and Stage II, LLM-based information extraction and hierarchical reconstruction. This architecture is operationalized through three detailed methodological stages: (1) automated BoM structure detection and preprocessing (Section 4.2.1), (2) dynamic schema generation and prompt construction (Section 4.2.2), and (3) LLM-based information extraction and hierarchical reconstruction into a structured foreground LCI-oriented representation (Section 4.3).

The approach builds on the principles of schema-driven prompting and structured information extraction discussed in Chapter 2 and draws specifically on the LMDX methodology (Perot et al., 2024), while adapting these principles to Excel-based industrial BoMs and to the specific objective of supporting foreground Life Cycle Inventory (LCI) generation. In the first high-level stage, deterministic preprocessing is used to convert heterogeneous Excel-based BoM files into a machine-processable representation. A Python-based preprocessing module automatically reads the Excel file, identifies the BoM header row based on the distribution of non-empty cells, extracts the corresponding column names, separates the BoM table from the metadata

region, and removes empty rows. The detected column structure is subsequently used to dynamically define the target output schema, ensuring that the attributes present in the original BoM are retained during extraction. In parallel, metadata located above the BoM table is preserved as a separate `bom_metadata` component. The second high-level stage uses an LLM to perform information extraction and hierarchical reconstruction. The processed BoM is converted into a textual representation and provided to the LLM through a structured prompt containing the BoM content, metadata, dynamically generated schema, and explicit extraction and hierarchy rules. The prompt instructs the model to preserve the detected entity attributes and reconstruct the product hierarchy according to the BoM Level attribute, whereby items with higher levels are nested under the nearest preceding item with a lower level, while items at the same level are treated as siblings. The resulting output is a machine-readable JSON representation containing BoM metadata and hierarchically organized BoM entities, which is subsequently consolidated as a structured intermediate representation for foreground LCI generation and other downstream LCA-related processes.

The architecture of the proposed methodology can be summarized as follows (Figure 4.1):

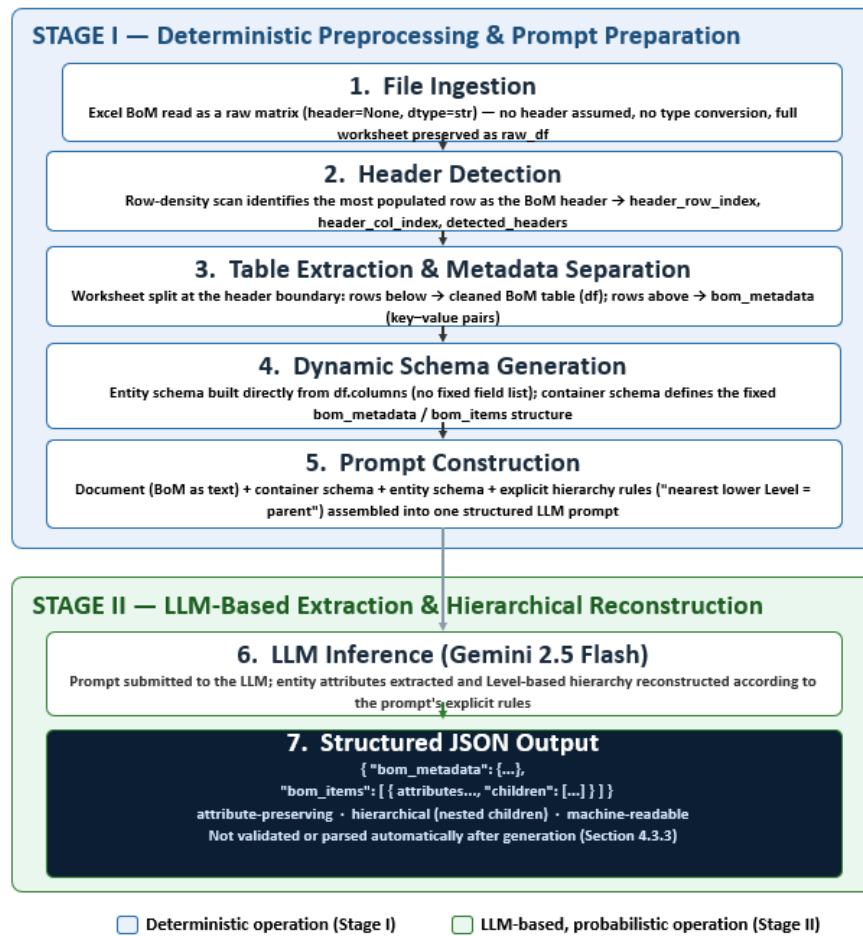


Figure 4.1 Proposed AI-Driven Methodology for Foreground LCI-Oriented Representation

Preservation of the product hierarchy is central to the proposed approach because the BoM hierarchy provides the structural relationships required to associate individual components with their respective assemblies and to maintain traceability when translating the product structure into a foreground LCI model. Within LCA, the foreground system is represented through interconnected unit processes, product flows, and elementary flows, and the consistent definition of these relationships is fundamental to the subsequent modelling and aggregation of environmental impacts (*ISO 14040:2006 - Environmental Management — Life Cycle Assessment — Principles and Framework*, n.d.). Consequently, explicitly preserving parent-child

relationships in the BoM supports the consistent association of components and assemblies during downstream LCI modelling. Industrial BoMs commonly encode these relationships through level indicators; however, their original tabular representation leaves the parent–child relationships implicit rather than explicitly represented as a hierarchical data structure. Furthermore, industrial BoM files may exhibit heterogeneous formatting, variable header locations, and metadata positioned differently across files, including among products from the same manufacturer. The proposed methodology therefore addresses both challenges by combining deterministic preprocessing to identify and organize the relevant BoM information with LLM-based reconstruction to transform the implicit hierarchy into an explicit, standardized, machine-readable representation suitable for subsequent foreground LCI generation. Integration with background LCA databases, such as Ecoinvent, is considered outside the scope of the current implementation and is discussed as a future extension, and the methodological and implementation limitations of the proposed approach are addressed in chapter 6.

The gap this methodology addresses is that no existing approach directly targets this combination of problem and input type. LMDX (Perot et al., 2024) addresses hierarchical entity extraction and localization from visually rich, unstructured documents, but has no equivalent to a pre-existing structural signal such as a Level attribute. Its hierarchical entities are recovered from table proximity and layout, not reconstructed from an explicit field the source document already provides. (Bejjit et al., 2026) apply a comparable hybrid deterministic-preprocessing-plus-constrained-LLM architecture to extract environmental and production metrics from unstructured company reports, but by explicit design produce a flat, atomic representation in which each extracted value is independent, the opposite output philosophy from the hierarchy-preserving representation this thesis requires. This research work

addresses precisely that remaining step: the reconstruction of hierarchical product structure from structured, but non-standardized, industrial BoMs, using an explicit structural signal (the Level attribute) that neither the document-understanding nor the flat-extraction literature has reason to exploit.

4.2 Stage I, Deterministic preprocessing and prompt preparation

4.2.1 BoM structure detection and preprocessing

- **File ingestion**

The preprocessing procedure begins with the direct ingestion of the Excel-based BoM using the Python programming language and the Pandas library, using the `read_excel` function through the following configuration:

```
raw_df = pd.read_excel(  
    file_name,  
    header=None,  
    dtype=str  
)
```

The `header=None` parameter is particularly important for the proposed methodology. In a conventional Excel data-processing workflow, the first row is often automatically interpreted as the table header. Such an assumption would be inappropriate for the input BoMs considered in this study because the main table may begin at a different row depending on the document structure. By disabling automatic header assignment, the complete worksheet is retained as a raw matrix, allowing the header-detection procedure to determine the most likely location of the BoM header based on the content of the worksheet rather than on a predefined position.

The `dtype=str` parameter is used to read the worksheet values as strings. This choice is intended to preserve the original textual representation of BoM information and to minimize unwanted type conversions during subsequent processing. Industrial BoMs may contain identifiers, part numbers, codes, descriptions, quantities, and other values that can be interpreted differently depending on their Excel formatting. In particular, identifiers containing leading zeros or alphanumeric combinations should not be unintentionally converted into numerical values. Treating the cells as strings therefore provides a consistent representation for the subsequent detection and extraction procedures.

Following ingestion, the resulting worksheet is stored as a Pandas DataFrame, referred to as `raw_df`. At this stage, the DataFrame does not yet represent the final BoM table. Instead, it constitutes an intermediate representation of the complete worksheet, including the metadata region, potential empty rows, the BoM header, and the BoM records. This distinction is important because the methodology deliberately postpones the definition of the final tabular structure until the header-detection procedure has been performed.

The ingestion stage can therefore be understood as a schema-independent loading step. No assumptions are initially made regarding the number of BoM attributes, the position of the header, or the starting column of the table. These characteristics are inferred in the subsequent preprocessing operations. The resulting `raw_df` consequently serves as the common input to the header-detection, metadata-extraction, and BoM-table-extraction procedures. This design establishes a clear separation between data ingestion and structural interpretation. The ingestion operation preserves the original worksheet content as completely as possible, while the subsequent stages determine which portions of that content correspond to metadata, headers, and BoM entities. This separation is particularly relevant for the objective of

developing a methodology that can be applied to heterogeneous industrial BoM files without requiring manual specification of the table location for each individual document.

- **Header detection**

Following raw Excel ingestion, the next step is to automatically identify the row containing the column headers of the BoM.

The implemented procedure is based on the number of non-empty cells contained in each row. The underlying assumption is that the BoM header generally represents the most structurally populated row within the region containing the main BoM table, since it contains the names of multiple attributes used to describe the subsequent entities. Each row of the raw DataFrame is therefore examined, for each row, the algorithm first excludes missing values, and excludes cells that are empty or contain only whitespace. This prevents visually empty cells from contributing to the estimation of the row density. The number of remaining cells is then used as the measure of row population. The header-detection algorithm evaluates the number of non-empty cells in each worksheet row and retains the row with the highest count as the candidate BoM header. The corresponding row and starting-column indices are recorded as `header_row_index` and `header_col_index`, respectively. The non-empty values from the detected starting column onward are then extracted as the BoM column headers.

The core implementation is based on the following procedure:

```
max_non_empty_count = 0
for row_idx, row in raw_df.iterrows():
    non_empty_cells = [
        str(cell).strip()
        for cell in row
        if pd.notna(cell) and str(cell).strip() != ""
    ]
```

```

if len(non_empty_cells) > max_non_empty_count:
    max_non_empty_count = len(non_empty_cells)
    header_row_index = row_idx
    header_col_index = row.first_valid_index()
    detected_headers = row.iloc[header_col_index:].tolist()

```

The output of this procedure consists of three principal pieces of structural information:

- `header_row_index`: the detected position of the BoM header
- `header_col_index`: the first populated column of the detected header
- `detected_headers`: the column names extracted from the detected header row.

Once the BoM header position has been identified, the worksheet is divided into two main regions: the metadata region preceding the header and the BoM table beginning at the detected header. The header position therefore defines the boundary between metadata and BoM records, while the detected starting column determines the portion of each subsequent row to be retained. The identified headers are then assigned to the extracted DataFrame, enabling the BoM to be represented as a conventional tabular dataset without requiring a predefined schema.

The header-detection procedure therefore serves as a key transition between the raw worksheet representation and the structured BoM representation used by the subsequent stages. Rather than imposing a fixed document layout, the methodology infers the table location from the content of each input file. This contributes to the adaptability of the preprocessing stage to BoMs with different header positions and varying numbers of attributes.

Now it is possible to consider a simplified worksheet structured as follows:

Row 0	Product	CCB54482BW
Row 1	Revision	01

Row 2					
Row 3	Level	Part No.	Description	Quantity	Material
Row 4	0	A001	Assembly	1	Steel
Row 5	1	A002	Component	2	Steel

The first two rows contain metadata and therefore have relatively few populated cells. Row 3 contains several populated cells corresponding to the BoM attributes and is consequently identified as the candidate header row. The detected header position then allows the worksheet to be divided into the metadata region above the header and the BoM table below it.

▪ **BoM Table extraction and cleaning**

Once the BoM header position has been identified, the worksheet can be divided into two main regions: the metadata region preceding the detected header and the BoM table beginning below the header.

Following the identification of the BoM header and, the rows constituting the actual BoM table are extracted from the raw worksheet. The purpose of this step is to transform the relevant portion of the Excel worksheet into a conventional tabular representation in which each row corresponds to a BoM entity and each column corresponds to an attribute of that entity. The extraction is performed using the header position and starting column identified during the automatic header-detection procedure.

The slicing operation starts at the row immediately following the detected header and extends to the end of the worksheet. At the same time, the extraction begins at the detected starting column and includes the subsequent columns. This ensures that the metadata region and any cells located before the beginning of the BoM table are excluded from the component-level dataset. The detected header values are subsequently assigned to the extracted table as its column names.

This operation converts the initial position-based representation of the worksheet into a named tabular structure. Importantly, the column names are not predefined in the implementation. They originate from the header row identified during preprocessing. Consequently, the resulting DataFrame reflects the attribute structure of the individual BoM being processed.

For example, if the detected header contains:

Level	Part No.	Description	Quantity	Material
-------	----------	-------------	----------	----------

The extracted DataFrame is structured accordingly:

Level	Part No.	Description	Quantity	Material
0	A001	Assembly	1	Steel
1	A002	Component	2	Steel

This dynamically assigned structure is important for the subsequent schema-generation stage because the column names provide the set of attributes that the LLM is instructed to preserve during extraction.

After the table has been extracted and its column names assigned, completely empty rows are removed. The algorithm removes a row only when all of its cells are missing. This distinction is important because partially populated rows are retained. A BoM may legitimately contain missing information for particular attributes, and removing such rows based on the presence of any missing value could result in the loss of valid component information. The cleaning operation is therefore intentionally limited to rows that contain no information at all.

At this point, the resulting DataFrame, `df`, represents the cleaned BoM table, with completely empty rows removed, that will be provided to the subsequent schema-generation and prompt-construction stages.

- **Metadata extraction**

The metadata region is extracted separately to ensure that document and product-level information is not lost when the worksheet is transformed into a structured BoM representation. In the proposed methodology, metadata is therefore treated as a distinct information layer from the component-level entities contained in the BoM table.

This operation deliberately does not impose a predefined set of metadata fields. Instead, the content of the region is examined dynamically, allowing the procedure to accommodate different types and arrangements of document-level information. This is particularly relevant for industrial BoMs, where information such as product identifiers, revisions, dates, document numbers, project information, or other contextual attributes may be present but may not occupy the same positions across files.

The extraction procedure processes the metadata region row by row. For each row, null cells and cells containing only whitespace are removed before determining how the remaining values should be represented. The implementation is based on the following logic:

```
bom_metadata = {}
for _, row in metadata_region.iterrows():
    cells = [
        str(cell).strip()
        for cell in row
        if pd.notna(cell) and str(cell).strip() != ""
    ]

    if not cells:
        continue

    if len(cells) == 1:
        bom_metadata[cells[0]] = ""
    elif len(cells) >= 2:
        key = cells[0]
```

```
value = " ".join(cells[1:])
bom_metadata[key] = value
```

For rows containing two or more non-empty cells, the first populated cell is interpreted as the metadata key, while the remaining cells are concatenated to form its corresponding value. For example, a row such as:

Product CCB54482BW

is represented as:

```
{
  "Product": "CCB54482BW"
}
```

Similarly, a row containing multiple value cells can be consolidated into a single metadata value. For example:

Revision 01 Final

would be represented as:

```
{
  "Revision": "01 Final"
}
```

This approach allows metadata information distributed across several worksheet cells to be converted into a simple key-value representation while retaining the textual content of the original cells.

Rows containing only one non-empty cell are handled differently. Since no separate value is available, the populated cell is treated as a metadata key and assigned an empty string as its value:

```
{
  "CONFIDENTIAL": ""
}
```

This rule allows standalone textual information to be retained rather than discarded. Such information may represent document labels, section markers, or other contextual information whose semantic role cannot be determined automatically from the worksheet alone. The methodology therefore preserves the information while avoiding the introduction of an unsupported interpretation.

This separation between `bom_metadata` and `bom_items` is intentional. Metadata describes the document, product, or BoM as a whole, whereas the entities contained in `bom_items` describe the individual components and assemblies that constitute the product. Maintaining these two information levels separately prevents document-level attributes from being incorrectly interpreted as component attributes during the subsequent LLM-based extraction process.

The metadata extraction procedure also contributes to the dynamic nature of the proposed methodology. Rather than requiring predefined metadata fields, it extracts the information available in each input document and incorporates it into the output structure. Consequently, the same preprocessing procedure can accommodate BoMs containing different types or numbers of metadata fields without requiring modifications to the extraction code.

4.2.2 Dynamic Schema Generation and Prompt Construction

Following the extraction and cleaning of the BoM table described in Section 4.2, the next stage defines the structure that will guide the LLM in representing the extracted information.

The dynamic schema is generated directly from the column names detected during the preprocessing stage. Since these column names originate from the header identified in the input worksheet, they represent the attributes

explicitly provided by the original BoM. The implementation creates an initially empty field for each detected column.

The resulting dictionary therefore contains one field for each attribute identified in the BoM. For example, if the extracted BoM contains the columns: Level, Part Number, Description, Quantity, Material. The generated entity schema takes the following conceptual form:

```
{
  "Level": "",
  "Part Number": "",
  "Description": "",
  "Quantity": "",
  "Material": ""
}
```

The empty values in this representation do not indicate missing information in the original BoM. Instead, they define the expected fields of an individual BoM entity and indicate where the corresponding values should be placed during the subsequent extraction process. The schema consequently functions as a structural template rather than as a dataset containing the final values. The actual values and hierarchical relationships are generated during the LLM-based extraction stage described in Section 4.4.

This dynamic approach differs from a fixed-schema strategy in which the developer would manually specify the attributes expected for every BoM. For example, a fixed schema might assume that every input contains Part Number, Description, and Quantity. Such an assumption could become restrictive when processing BoMs containing additional manufacturer-specific attributes or alternative naming conventions. By deriving the schema from `df.columns`, the proposed methodology adapts the entity representation to the attributes actually present in each input document.

- **Container schema**

Following the generation of the dynamic entity schema, the methodology defines a higher-level container schema that organizes the information extracted from the BoM. While the dynamic entity schema specifies the attributes associated with individual BoM entities, the container schema defines the overall structure in which these entities and the document-level metadata are represented.

The container schema is generated programmatically using the metadata extracted during preprocessing and an initially empty `bom_items` collection:

```
container_structure = f"""
{{
  "bom_metadata": {json.dumps(bom_metadata, indent=2)},
  "bom_items": []
}}"""
```

The resulting structure is:

```
{
  "bom_metadata": {},
  "bom_items": []
}
```

The `bom_metadata` object contains information extracted from the worksheet region preceding the detected BoM header, such as product identifiers, revisions, or other document-level attributes. In contrast, `bom_items` is an array reserved for the BoM entities extracted from the main table. At this stage, the array remains empty because entity extraction and hierarchical reconstruction are performed subsequently.

As with the entity schema, the container schema's structure reflects this dynamic principle, but only partially. Its top-level organization, `bom_metadata` and `bom_items`, is defined by the methodology, whereas the content of `bom_metadata` and the attributes of individual BoM entities depend on the input document and the dynamically generated entity schema. This

combination provides a consistent outer representation while accommodating variations in BoM structure and attributes.

Consequently, the container schema establishes the structural interface between preprocessing and LLM-based extraction, preserving document-level metadata while providing the framework into which extracted entities and their hierarchical relationships are subsequently incorporated.

The distinction between the two schema levels is summarized visually in Figure 4.1 and compared directly in Table 4.1 below. Figure 4.1 contrasts the dynamically generated entity-level fields against the fixed top-level container structure, while Table 4.1 summarizes their differences in terms of purpose, source, level, and variability

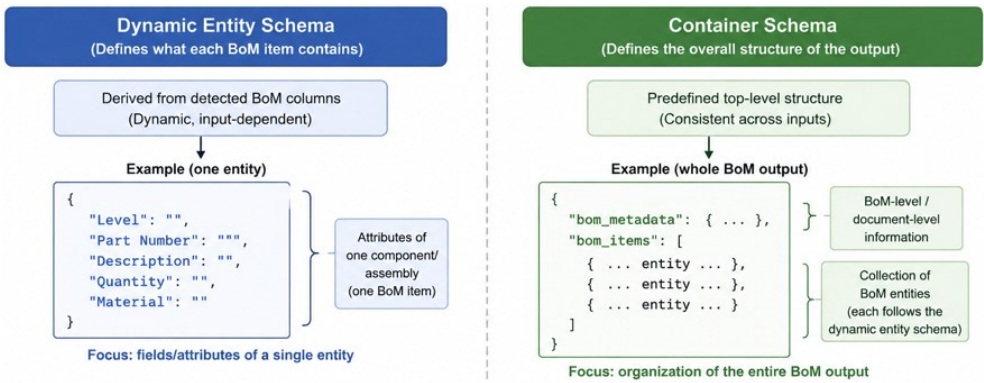


Figure 4.2 Structure Differences between Dynamic Entity Schema and Container Schema

Table 4.1 Key Differences between Dynamic Entity Schema and Container Schema

	Dynamic Entity Schema	Container Schema
Purpose	Defines the attributes/fields of each BoM entity	Defines the overall structure of the complete BoM structure
Source	Generated dynamically from detected column names	Predefined by the methodology (top-level structure)
Level	Entity level	Document level

Example	“Level”, “Part number”, “Description”	“bom_metadata”, “bom_items” (which contains entities)
Variability	Varying across different BoMs	Top-level structure is fixed, contents are dynamic

▪ Hierarchy Rules as Prompt Constraints

A central requirement of the proposed methodology is the preservation of the hierarchical structure encoded in the original BoM. Although the BoM is provided as a flat tabular structure, its Level attribute provides an explicit structural indicator of relationships between products, assemblies, sub-assemblies, and components. The methodology therefore constrains the LLM to use this attribute when transforming the tabular representation into a nested JSON hierarchy, rather than inferring parent–child relationships solely from component descriptions or part numbers. In this way, the LLM performs a constrained structural transformation of the source BoM, while the hierarchy rules defined in the prompt specify how the existing Level information must be interpreted.

The hierarchy rules are incorporated directly into the prompt as follows:

The hierarchy is strictly based on the "Level" attribute:

- *Higher "Level" values must be nested under the nearest row with a lower "Level".*
- *Same "Level" values are siblings within the same parent.*

These instructions define how the sequence of rows in the original flat table should be transformed into parent–child relationships. Two complementary principles define this transformation: a parent-assignment rule and a sibling rule.

Parent-assignment rule: An entity is assigned as the child of the nearest preceding entity whose Level value is strictly lower than its own. "Nearest

preceding" is essential: the parent of a given row is not necessarily the row immediately above it, but the closest prior row satisfying the lower-Level condition, a distinction that becomes critical once the hierarchy spans several consecutive depths.

Sibling rule: Entities sharing the same Level value, and occurring consecutively under the same unresolved parent context, are treated as siblings rather than as being nested within one another.

Together, these rules allow the depth of the hierarchy to expand or contract at any point in the sequence: a row may descend to a deeper Level than its predecessor, return to a shallower one, or remain at the same Level, and the two rules above are formulated to resolve each case unambiguously.

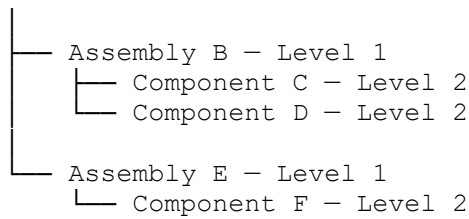
It is important to note that these rules are not enforced through deterministic code. They are formulated as natural-language constraints embedded in the LLM prompt, and their correct application depends entirely on the model's interpretation of the instruction rather than on a verified algorithmic procedure. The formalization presented here, the parent-assignment and sibling rules, describes the intended interpretation that the prompt is designed to elicit, not a guarantee that is programmatically checked or enforced.

The following example illustrates the interpretation the two rules are intended to produce, not a step executed by the implementation.

Row	Level	Entity
1	0	Product A
2	1	Assembly B
3	2	Component C
4	2	Component D
5	1	Assembly E
6	2	Component F

Applying the two rules yields:

Product A — Level 0



Component C is assigned to Assembly B by the parent-assignment rule, as Assembly B is the nearest preceding row with a lower Level. Component D, occurring at the same Level as Component C, is treated as its sibling rather than as its child, an application of the sibling rule that keeps both under Assembly B. When Assembly E subsequently appears at Level 1, the parent-assignment rule again applies with respect to Product A (Level 0), placing Assembly E as a sibling of Assembly B rather than a descendant of either B, C, or D. This transition from Level 2 back to Level 1, demonstrates why the parent must be identified by nearest lower Level rather than by positional adjacency: Assembly E's nearest preceding lower-Level row is Product A, not Component D. Component F, following at Level 2, is then assigned to Assembly E by the same logic.

▪ Relationship with the target JSON structure

The hierarchy constraints determine how the bom_items collection generated according to the container schema is populated. The dynamic entity schema specifies the attributes associated with each entity, while the hierarchy rules determine how those entities are structurally related.

For example:

```

{
  "bom_items": [
    {
      "Level": "0",
      "Part Number": "A",
      "Description": "Product",
      "children": [
        {
          "Level": "1",

```

```

    "Part Number": "B",
    "Description": "Assembly",
    "children": [
      {
        "Level": "2",
        "Part Number": "C",
        "Description": "Component"
      }
    ]
  }
]
}

```

Here, the entity schema determines fields such as Level, Part Number, and Description, whereas the hierarchy constraint determines that C is nested under B, and B is nested under A.

This distinction is important because correct attribute extraction alone would not be sufficient for the intended LCI application. A flat representation could preserve all component information while still losing the relationships between assemblies and their constituent components. The hierarchy constraint therefore ensures that the structural relationships encoded in the original BoM are explicitly represented in the resulting machine-readable output.

▪ Prompt Construction

Following the definition of the dynamic entity schema, container schema, and hierarchy constraints, these elements are combined with the processed BoM content to construct the final prompt submitted to the Large Language Model (LLM). The prompt is constructed programmatically using Python string interpolation. The resulting prompt also includes explicit instructions governing the reconstruction of the BoM hierarchy.

The purpose of this step is to create a single structured instruction that provides the model with both the source information to be extracted and the constraints governing its representation. The prompt therefore constitutes the interface between the deterministic preprocessing and schema-definition stages and the subsequent LLM-based extraction stage.

- **BoM content as the extraction document**

The cleaned BoM DataFrame obtained during preprocessing is first converted into a textual representation. This operation converts the structured DataFrame into a plain-text table while excluding the Pandas index. The resulting representation contains the detected column names and the corresponding BoM values and provides the source content that the LLM must process. The table is then explicitly delimited within a <Document> section:

<Document>

[BoM table]

</Document>

The use of an explicit document boundary separates the source data from the instructions that follow. This organization is intended to make clear which information constitutes the input document and which text constitutes the extraction task.

For example, a simplified input to the prompt may take the following form:

<Document>

Level	Part Number	Description	Quantity	Material
0	A001	Main Assembly	1	Steel
1	A002	Component	1	Steel

</Document>

The textual conversion does not introduce a new BoM schema or alter the hierarchical information contained in the source table. Rather, it provides a representation of the already processed BoM that can be included in the LLM request.

- **Integration of the target structure**

The dynamically generated container schema is subsequently inserted into the prompt to communicate the required organization of the output. The relevant part of the implementation is:

```
prompt = f"""
```

```
<Document>
```

```
{table_text}
```

```
</Document>
```

```
<Task>
```

Extract the text values and tags of the entities from the document and return a properly formatted, hierarchically nested JSON object.

The output must follow this overall structure:

```
{container_structure}
```

The variable container_structure contains the top-level organization defined previously:

```
{
```

```
  "bom_metadata": {},
```

```
  "bom_items": []
```

```
}
```

This explicitly instructs the LLM that the output must distinguish between information concerning the BoM or product as a whole and the individual entities extracted from the BoM table.

The dynamic entity schema is then included separately in the task instructions:

Each entity inside "bom_items" must include the following fields:

{entity_fields}

Here, entity_fields is generated from the detected columns of the input BoM. Consequently, the final prompt communicates both the overall output structure and the attribute structure expected for each entity.

- **Integration of hierarchy constraints**

The prompt then incorporates the hierarchy rules defined previously.

These instructions transform the prompt from a simple information-extraction request into a structure-constrained extraction task. The LLM is not only required to identify and reproduce the information contained in the BoM, but also to represent the relationships encoded by the Level attribute in nested form.

The resulting prompt can therefore be conceptualized as consisting of four functional components:

Table 4.2 Functional Components of the LLM Extraction Prompt

Prompt component	Source	Function
Document	Cleaned df DataFrame	Provides the BoM information to be extracted
Container schema	container_structure	Defines the organization of the complete JSON output
Entity schema	entity_fields	Defines the attributes to be preserved for each BoM entity
Hierarchy constraints	Explicit prompt instructions	Defines how parent–child and sibling relationships are reconstructed

This organization provides a clear separation between data, output structure, and processing instructions.

- **Programmatic assembly**

The complete prompt is assembled through an f-string, allowing the elements generated for each individual BoM to be inserted automatically:

```
prompt = f"""
```

```
<Document>
```

```
{table_text}
```

```
</Document>
```

```
<Task>
```

Extract the text values and tags of the entities from the document and return a properly formatted, hierarchically nested JSON object.

The output must follow this overall structure:

{container_structure}

Each entity inside "bom_items" must include the following fields:

{entity_fields}

The hierarchy is strictly based on the "Level" attribute:

- Higher "Level" values must be nested under the nearest row with a lower "Level".

- Same "Level" values are siblings within the same parent.

Flat structures, incorrect nesting, or malformed JSON are unacceptable.

</Task>

<Extraction>

""

The same dynamic property extends to the prompt itself, which is assembled rather than fixed:

- table_text changes according to the content of the BoM
- entity_fields changes according to the detected BoM columns
- container_structure incorporates the metadata extracted from the specific document;
- Hierarchy rules remain methodologically consistent across inputs.

▪ **Role of prompt construction in the proposed methodology**

The prompt construction stage is therefore not considered an independent information-extraction operation. Rather, it is the integration stage that prepares the complete LLM input. Its role is to translate the outputs of the

deterministic preprocessing module into a structured instruction that constrains the subsequent probabilistic processing.

This design also provides a degree of reproducibility. For a given BoM, the preprocessing procedures generate the same detected table structure and associated schema, and these outputs are inserted into a consistent prompt template. The LLM consequently receives a standardized instruction format while the document-specific content and attributes remain dynamic.

4.3 Stage II, LLM-Based Extraction and Hierarchical Reconstruction

4.3.1 LLM Selection, Inference, and Extraction Procedure

Following the deterministic preprocessing and dynamic prompt construction described in Sections 4.2.1 and 4.2.2, the third stage of the proposed methodology consists of LLM-based information extraction and hierarchical reconstruction. At this stage, the BoM-specific structured prompt generated previously is submitted to a Large Language Model (LLM), which is responsible for extracting the entity information from the processed BoM and representing the implicit product hierarchy as an explicit nested structure.

- **LLM Selection and Inference**

The proposed implementation uses Gemini 2.5 Flash as the LLM responsible for information extraction and hierarchical reconstruction. The model is accessed programmatically through the Google Generative AI interface, with the API receiving the complete BoM-specific prompt constructed during Stage 2. This prompt contains the processed BoM content, the detected entity attributes, the target output structure, the extracted metadata, and the explicit hierarchy constraints. For each input BoM, the same general inference

procedure is applied, while the content of the prompt is adapted to the structure and attributes detected in the corresponding document.

Gemini 2.5 Flash was adopted directly, for practical reasons, rather than selected after a formal comparative evaluation against alternative models (e.g., GPT-4-class models or the larger Gemini Pro tier), no such comparison was performed as part of this work. The choice reflects the practical characteristics generally associated with "Flash"-tier models relative to their "Pro"-tier counterparts within the same model family: lower inference latency and lower per-request cost, which are relevant for a preprocessing pipeline intended to process a batch of BoM files rather than a single interactive query, together with a context window sufficient to accommodate the textual representation of the BoM tables used in this work. A systematic comparison of candidate LLMs for this specific extraction and hierarchy-reconstruction task was not carried out and is identified as a direction for future evaluation.

The use of an LLM at this stage is motivated by the need to process heterogeneous industrial BoMs while maintaining a flexible interface between the source representation and the target structured representation. The preceding Python-based preprocessing stage handles variations that can be addressed through deterministic rules, including differences in header position, column number, metadata location, table length, and empty-row structure. However, purely deterministic processing becomes increasingly dependent on predefined rules when variations involve attribute terminology, textual representations, or document-specific conventions. Unlike traditional machine learning approaches, which typically require a large, task-specific labeled dataset before they can be applied reliably, a pre-trained LLM can be used directly, without requiring a dedicated training dataset of annotated industrial BoMs. The LLM therefore provides a flexible information-extraction layer capable of interpreting the processed BoM within the dynamically generated schema and the constraints defined by the

methodology. Importantly, LLM inference does not determine the schema or the fundamental hierarchy rules. The schema is generated deterministically from the attributes detected during preprocessing, while the hierarchy rules are explicitly specified in the prompt. The role of the LLM is consequently limited to extracting and organizing the BoM information according to these predefined structural constraints. This separation between deterministic processing and LLM inference combines the reproducibility and control of programmatic operations, including the prompt-level reproducibility discussed in Section 4.2.2, with the flexibility of model-based information extraction.

The approach also allows a common extraction framework to be applied across BoMs with different column configurations without requiring a separate extraction procedure for each document format. Nevertheless, LLM inference introduces probabilistic behavior and may produce errors such as missing attributes, incorrect values, malformed JSON, or inappropriate hierarchical relationships. The schema and hierarchy constraints are therefore used to reduce the model's output variability, while the resulting representation is subsequently evaluated during the validation stage.

4.3.2 Expected LLM Output and target representation

The expected output of the LLM is a machine-readable JSON representation containing both the extracted BoM metadata and the hierarchically organized BoM entities.

Each entity is expected to preserve the attributes defined by the dynamically generated entity schema. In addition, hierarchical relationships are represented through nested children elements where applicable.

The expected output consequently has three important characteristics:

- First, it is machine-readable. JSON is used because it provides an explicit and programmatically accessible representation of both attributes and relationships.
- Second, it is attribute-preserving. The entity fields correspond to the attributes detected in the source BoM rather than to an entirely predefined universal schema.
- Third, it is hierarchical. The children relationships explicitly represent the product structure encoded by the original Level attribute.

The expected output is therefore not intended to replace the original BoM but to provide a structured intermediate representation that can subsequently support automated foreground LCI generation.

It is also important to distinguish between the expected output structure and the actual model response. The methodology defines the former through the prompt, but the LLM may produce a response that does not completely conform to the required structure. Such deviations constitute potential error conditions and are considered below.

This target structure is the representation the LLM is instructed to return directly. The current implementation does not perform any additional parsing, validation, or restructuring of the model's response after generation. The raw text output is expected to already conform to this structure by construction of the prompt, but is not programmatically verified against it. The requested representation is a hierarchical JSON object containing two main components: `bom_metadata`, which preserves information associated with the BoM or product as a whole, and `bom_items`, which contains the individual BoM entities together with their attributes and reconstructed parent-child relationships.

A simplified example is shown below:

```

{
  "bom_metadata": {
    "Product": "CCB54482BW",
    "Revision": "03"
  },
  "bom_items": [
    {
      "Level": "0",
      "Part Number": "A001",
      "Description": "Main Assembly",
      "Quantity": "1",
      "Material": "Steel",
      "children": [
        {
          "Level": "1",
          "Part Number": "A002",
          "Description": "Component",
          "Quantity": "1",
          "Material": "Steel"
        }
      ]
    }
  ]
}

```

JSON was selected as the target representation because it provides a flexible and machine-readable structure capable of representing both entity attributes and nested product relationships. Its compatibility with programmatic processing also facilitates the subsequent transformation of BoM information into foreground LCI data. In this context, the JSON representation should be understood as an intermediate data layer between the industrial BoM and subsequent LCI modelling, rather than as an LCI dataset itself.

The complete methodology is therefore designed to terminate with a structured representation that preserves three principal aspects of the source information: BoM-level metadata, entity-level attributes, and product hierarchy. This representation can subsequently support the identification and organization of foreground inventory elements, including quantities, materials, and component relationships.

4.3.3 Error Conditions and Robustness

As the extraction relies on LLM inference, potential errors may include malformed JSON, missing or altered attributes, omitted entities, or incorrect hierarchical relationships. The methodology addresses these risks primarily through explicit schema and hierarchy constraints incorporated into the prompt, which restrict the expected output structure and guide the model toward preserving the source BoM information. However, these constraints do not guarantee error-free generation, and the current implementation does not include an automated validation and correction loop. The accuracy and robustness of the generated hierarchical representation are therefore assessed during the subsequent validation stage.

5. VALIDATION of the PROPOSED AI-ASSISTED METHODOLOGY

5.1 Introduction

Chapter 4 presented the proposed AI-assisted methodology for automatically transforming heterogeneous industrial Bills of Materials (BoMs) into structured, hierarchical, and machine-readable representations suitable for foreground Life Cycle Inventory (LCI) development. This chapter presents the validation of that methodology, following a two-stage approach.

The first stage, presented in Section 5.2, consists of a technical validation of the pipeline's core functionalities, combining a qualitative and a quantitative assessment. The second stage, presented in Section 5.3, consists of an industrial case study applying the proposed methodology to a single representative product, a combination oven–microwave, and comparing the automatically generated structured representation against a reference foreground inventory developed through conventional manual disassembly. This stage evaluates the practical applicability and contribution of the methodology within a real foreground LCI development workflow.

Section 5.4 discusses the implications of both validation stages for the practical use of the methodology in supporting foreground LCI preparation and identifies the limitations that define the current scope of the work.

5.2 Technical validation

The technical validation of the proposed methodology is conducted through two complementary forms of assessment: a qualitative validation and a quantitative validation. The qualitative validation, presented in Section 5.2.2, is used to verify the overall functioning of the methodology by checking

whether each of its core processing stages behaved correctly for a given BoM, namely whether the BoM table and its header were correctly detected, whether the associated bom_metadata was correctly identified and separated from the bom_items, and whether the extracted information was successfully organized into the structured JSON representation defined in Chapter 4. This assessment therefore provides a stage-by-stage view of whether the pipeline completed each step of the process as intended.

The quantitative validation, presented in Section 5.2.2, goes further by examining the accuracy of the generated output at a more granular level. Rather than assessing whether a processing stage succeeded or failed as a whole, it focuses on whether each individual element of the BoM, namely each extracted component, its associated attributes, and its position within the reconstructed product hierarchy, was correctly represented in the generated output. This assessment is conducted by comparing the generated JSON against a manually verified reference structure for each BoM and computing standard information-extraction metrics, namely Precision, Recall, and F1-score.

The validation dataset consists of two complementary categories of BoMs, summarized in Table 5.1. The first category comprises real industrial BoMs obtained from different manufacturing companies, exhibiting differences in document organization, hierarchy depth, table positioning, and component arrangement; these provide realistic test cases for evaluating the methodology's ability to process engineering documents without manual adaptation.

The second category consists of controlled synthetic BoMs developed specifically for this research, not to simulate industrial products but to reproduce particular structural situations, such as a BoM whose inventory table does not begin at the top of the worksheet, or one containing metadata

preceding the table, that challenge specific stages of the processing pipeline. These controlled examples allow individual components of the methodology, particularly header detection, table localization, and data parsing, to be evaluated under conditions not consistently available in the industrial dataset.

This combination of qualitative and quantitative assessment across a deliberately heterogeneous dataset allows the technical robustness of the proposed methodology to be evaluated both in terms of whether it functions correctly and in terms of how accurately it performs.

Table 5.1 Characteristics of the technical evaluation datasets

Dataset Type	Main characteristic	Validation objective
BoM 1 Synthetic	Simplified structure	General validation
BoM 2 Synthetic	Offset table position, metadata preceding table	Document flexibility
BoM 3 Industrial	Complex hierarchy (dot-notation, up to 9 levels)	Hierarchy reconstruction
BoM 4 Industrial	Hierarchy defined through keyword ("assieme") rather than a level indicator	Structural robustness
BoM 5 Industrial	Large BoM (360 components, hierarchy depth to level 6)	Scalability
BoM 6 Industrial	Flat structure, no hierarchy indicator, different formatting and organization	Robustness evaluation

5.2.1 Qualitative validation

The qualitative validation was conducted through manual visual comparison between each generated JSON output and its corresponding original BoM: for each dataset, the researcher verified whether the BoM table and header

had been correctly located, whether the metadata region had been correctly identified and separated from the component list, and whether the corresponding JSON output had been successfully generated, following the structure defined in Chapter 4. The correctness of the extracted components, their attributes, and the reconstructed hierarchy is assessed separately, through the quantitative validation presented in Section 5.2.4

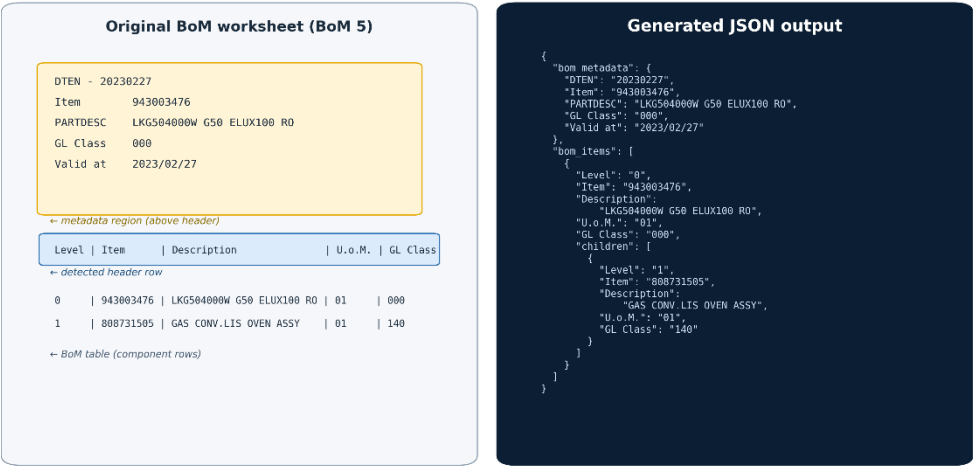


Figure 5.1 Correspondence between the original BoM worksheet and the generated JSON output for BoM 5, illustrating metadata separation, header detection, and hierarchical structuring.



Figure 5.2 Correspondence between the original BoM worksheet and the generated JSON output for BoM 6, illustrating metadata handling and the flat structure produced for a BoM with no explicit hierarchy indicator.

Figures 5.1 and 5.2 illustrate the same correspondence between worksheet and JSON output for two of the evaluated industrial BoMs. In BoM 5 (Figure 5.1), the metadata region preceding the header (item identifier, description, GL class, and validity date) is correctly separated into `bom_metadata`, while the detected header row defines the entity schema used to structure the component rows into `bom_items`, with the parent–child relationship between the root item and its first-level assembly preserved through nesting. In BoM6 (Figure 5.2), the metadata region consists of a single descriptive line rather than a set of key-value pairs, and is accordingly represented as a standalone key with an empty value in `bom_metadata`, consistent with the metadata-extraction rule defined in Chapter 4; the extracted components are represented as a flat list within `bom_items`, without nested children, reflecting the absence of a hierarchy indicator in the source BoM.

The same qualitative validation procedure described above and illustrated in Figures 5.X and 5.Y for BoM 5 and BoM 6, was applied to each of the remaining BoMs in the technical validation dataset. For each dataset, the metadata region, the BoM header, and the resulting JSON output were manually compared against the original worksheet following the criteria described in Section 5.2.2. Results across all six evaluated BoMs are summarized in Table 5.2.

Table 5.2 Qualitative validation results across the evaluated BoMs

BoM	<code>bom_metadata</code> detected	<code>bom_items</code> / header detection	JSON extracted
BoM 1	✓	✓	✓
BoM 2	✓	✓	✓
BoM 3	✓	✓	✓
BoM 4	✓	✓	✓
BoM 5	✓	✓	Partial
BoM 6	✓	✓	✓

As shown in Table 5.2, the qualitative validation confirms that BoM and header detection, and metadata separation, succeeded for all six evaluated BoMs, regardless of formatting differences such as offset table positions, varying hierarchy encodings, or the presence and structure of metadata preceding the table. This indicates that the deterministic preprocessing stage of the pipeline is robust across the range of document structures represented in the dataset.

JSON generation succeeded fully for five of the six BoMs. For BoM 5, JSON generation is recorded as partial, at 360 components and six hierarchy levels, the single-pass processing constraint affects the pipeline's ability to reliably complete the full structured output. This result is examined in more depth in the quantitative validation (Section 5.2.2), where Precision, Recall, and F1-score isolate which specific aspects of the output, entity extraction versus hierarchy reconstruction, are affected at this scale.

5.2.2 Quantitative validation

While the qualitative validation described in Section 5.2.1 verifies whether each processing stage of the pipeline completed successfully, it does not measure how accurately the content of the generated output reflects the source BoM. To address this, a quantitative validation was conducted, evaluating the pipeline's performance at two levels: the extraction of individual BoM components, and the reconstruction of the product hierarchy. This evaluation was conducted using (i) Precision, the proportion of generated elements that are correct, (ii) Recall, the proportion of true elements that were correctly identified, and (iii) F1-score, the harmonic mean of Precision and Recall, providing a single balanced measure of both. These three metrics were computed from True Positive (TP), False Positive (FP), and False Negative (FN) counts: a True Positive is an element present in both the generated output and the reference structure; a False Positive is an element present in

the generated output but absent from the reference, indicating a spurious or incorrect result, and a False Negative is an element present in the reference but missing from the generated output, indicating an omission. This TP/FP/FN framework, together with the corresponding Precision, Recall, and F1-score formulas, follows the standard evaluation approach used in information extraction and relation extraction tasks, and is applied here at both the extraction level and the hierarchy-reconstruction level, with the specific definition of what constitutes an element, an attribute value or a parent-child pair, adapted to each level as described below

- **Reference structure and comparison procedure**

For each BoM, a reference structure was manually constructed from the original spreadsheet, representing the expected components, attributes, and hierarchical relationships. The JSON output generated by the proposed methodology was converted into a flat tabular representation, with one row corresponding to each extracted component, allowing direct row-by-row comparison against the reference. This flattening step was necessary because the generated output is a nested structure, while the comparison logic operates on tabular data.

- **Extraction validation**

Component-level correctness was assessed by comparing, for each attribute field, the value present in the reference against the corresponding value in the generated output. For each reference cell, the formula “=IF(A7="", "", IF(COUNTIF(\$AD:\$AN, A7) > 0, "TP", "FN"))” was used to determine whether the value was present somewhere in the generated columns (True Positive) or missing from the generated output (False Negative). Conversely, the formula “=IF(AD8="", "", IF(COUNTIF(\$A:\$I, AD8) > 0, "TP", "FP"))” was applied to the generated values to identify values correctly present in the reference (TP)

or introduced without a corresponding reference value (False Positive). Precision, Recall, and F1-score were then computed as:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

It should be noted that, as implemented, this check was applied across all extracted attribute fields for each component (e.g. part number, description, quantity, unit of measure, material, weight, region, and lifecycle phase) rather than to the component identifier alone. The resulting Precision, Recall, and F1-score therefore represent a combined measure of entity identification and attribute correctness, rather than a metric isolating entity identification alone.

▪ **Hierarchy validation**

Hierarchical correctness was evaluated separately, since a component can be correctly identified while still being incorrectly positioned within the product structure. Rather than evaluating item presence alone, this validation targeted the relational integrity of the hierarchy, whether each component was correctly linked to its immediate parent, since this relationship constitutes the core structural information required for accurate foreground LCI generation. For each component in the reference BoM, its parent was first determined as the nearest preceding row whose hierarchy level was one unit lower, using the formula `=IF(B3=0,"",XLOOKUP(B3-1,$B$2:B2,$C$2:C2,"",0,-1))`, which returns no parent for root-level (Level 0) components and otherwise searches backward through the Level column to locate the nearest ancestor,

directly mirroring the parent-assignment rule specified in the extraction prompt. Using the resulting item and parent columns, each reference component was checked against the generated structure by testing, through a COUNTIFS condition, whether a row existed in the generated output with both the same item and the same parent: a match was classified as a True Positive, and its absence as a False Negative, the equivalent check performed in the opposite direction, testing whether each generated (item, parent) combination existed in the reference, classified matches as True Positives and unmatched generated pairs as False Positives, indicating a spurious or incorrectly attached relationship. A true negative was not defined, since the space of non-existent parent-child combinations within a hierarchical structure is unbounded and therefore not a meaningful comparison category, a limitation inherent to the evaluation of tree and graph structures more broadly. Precision, Recall, and F1-score were computed using the same formulas as above, now applied to (item, parent) matches rather than individual attribute values.

- **Level accuracy (secondary hierarchy check)**

To support this check, each matched (item, parent) pair was additionally represented as a single concatenated key, formed as “item|parent”, allowing the corresponding row in the reference structure to be located directly for each confirmed match. In addition to the pair-level comparison, a depth-consistency check was performed on the subset of pairs already confirmed as True Positives, verifying whether the matched component was recorded at the same hierarchical level in both the reference and generated structures. This check is not a precision/recall measure in itself, since it applies only to pairs already confirmed correct and has no independent false-positive or false-negative concept: a True Positive pair is either at the correct depth or not. It is instead reported as a simple conditional accuracy, Level Accuracy = OK /

(OK + Mismatch), providing a secondary validation dimension beyond relational correctness alone.

- **Rationale for the chosen metrics**

Precision, Recall, and F1-score were selected over a simple accuracy measure because the extraction and reconstruction tasks performed by the pipeline can fail in two structurally different ways: components, attributes, or relationships present in the reference may be omitted from the generated output (false negatives), while elements not present in the reference may be erroneously introduced (false positives). Precision and Recall explicitly separate these two failure modes, quantifying respectively the correctness and the completeness of the generated output, while the F1-score provides a single harmonic-mean measure balancing both. This framework is a standard evaluation approach for information extraction and relation extraction tasks (Tjong et al., n.d.), in which entities and their relationships must be identified from unstructured or semi-structured sources, a problem structurally analogous to the extraction of components and hierarchical relationships from industrial BoMs addressed in this work. The same framework was subsequently applied to the evaluation of hierarchical reconstruction, using parent–child relationships rather than individual components as the unit of comparison, which represents a natural and established extension of the metric to relation-level evaluation.

A. Extraction validation

Component 943005514 (Level 0) — reference value checked against generated output, field by field

#	Level	Item	Description	U.o.M.	Item tp.	GL Class
1	0	943005514	CC85442CBM..	01	22	000
TP	TP	FN	FN	TP	TP	TP

→ 2 of 9 checked fields not found in the generated output → counted as False Negatives

B. Hierarchy validation

(item, parent) pairs — reference structure vs. generated structure

Reference pair: 808731704 808731714	Generated pair: 808731704 808731714
Identical in both → True Positive (TP)	
Reference pair: 342486205 943005514	Generated pair: 342486205 (none)
not the same pair on both sides → one False Negative (reference) + one False Positive (generated)	

(all TP pairs for this BoM were additionally found to be at the correct depth in the hierarchy)

Figure 5.3 Worked example of the quantitative validation procedure applied to BoM 1: extraction validation (Panel A) and hierarchy validation (Panel B).

Figure 5.3 illustrates this procedure applied to BoM 1. In the extraction validation (Panel A), component 943005514 (the Level 0 root item) is checked field by field: seven of its nine attribute values, including Level, U.o.M., Item tp., and GL Class, are correctly matched in the generated output, while the Item and Description fields are not found matching and are accordingly counted as False Negatives. In the hierarchy validation (Panel B), the pair 808731704 | 808731714 appears identically in both the reference and generated structures and is therefore counted as a True Positive, whereas the reference pair 342486205 | 943005514 has no counterpart in the generated output, component 342486205 is instead recorded there with no parent at all, producing one False Negative on the reference side and one False Positive on the generated side.

The same procedure was applied to the remaining five BoMs. Table 5.3 presents the resulting Precision, Recall, and F1-score for extraction and hierarchy validation across all six BoMs.

Table 5.3 Quantitative validation results: Precision, Recall, F1-score, and Level Accuracy for extraction and hierarchy reconstruction across the evaluated BoMs

BoM	Extraction Precision	Extraction Recall	Extraction F1	Hierarchy Precision	Hierarchy Recall	Hierarchy F1	Level Accuracy
BoM 1	100%	99,81%	99,90%	58,20%	57,72%	57,96%	100%
BoM 2	100%	100%	100%	100%	100%	100%	100%
BoM 3	99,85%	99,85%	99,85%	100%	100%	100%	100%
BoM 4	99,76%	98,06%	98,41%	N/A	N/A	N/A	N/A
BoM 5	37,12%	59,03%	45,58%	59,29%	23,06%	33,20%	100%
BoM 6	95,23%	96,09%	95,66%	00,00%	undefined	undefined	N/A

5.2.3 Discussion of technical validation

The qualitative and quantitative validations were designed to be complementary: the qualitative assessment verifies whether each processing stage completed at all, while the quantitative assessment measures how accurately the resulting structure reflects the source BoM. Comparing the two across the six evaluated BoMs reveals both agreement and, in some cases, informative disagreement.

BoM 5 was already flagged as a partial result in the qualitative assessment, on account of its scale (360 components, hierarchy depth to level 6). The

quantitative results confirm this: extraction Precision and Recall are markedly lower for BoM 5 (37.12% and 59.03% respectively) than for any other evaluated BoM, and hierarchy Recall falls to 23.06%, meaning fewer than a quarter of the true parent-child relationships in this BoM were correctly reproduced. However, the Level Accuracy result offers an important qualification to this finding: among the parent-child pairs that were correctly reconstructed, all were also positioned at the correct depth within the hierarchy (Level Accuracy = 100%). This indicates that the pipeline's difficulty with BoM 5 lies specifically in identifying the correct parent-child relationships at this scale, rather than in misplacing components once a relationship has been correctly captured, a distinction consistent with the single-pass, non-chunked processing strategy discussed in Chapter 4, where the primary risk as document size increases is that relationships are missed or fragmented across the single model call, not that captured relationships are internally inconsistent.

BoM 1 illustrates a case where the qualitative and quantitative assessments disagree, and the disagreement is informative. BoM 1 was recorded as a qualitative success, correct header detection, correct metadata separation, well-formed JSON. The quantitative results confirm this at the extraction level (F1 = 99.90%) but reveal a substantially weaker hierarchy score (F1 = 57.96%, though again with a perfect Level Accuracy of 100% among the pairs correctly matched) that the qualitative, stage-level assessment could not detect. Closer inspection traces both results to a single point of failure: the root-level component (Level 0) was extracted with a malformed identifier in the generated output, and the 51 components that should have been nested beneath it in the reference structure instead appear in the generated JSON with no parent recorded at all, accounting for 51 of the 52 hierarchy False Negatives and all 51 hierarchy False Positives for this BoM. This demonstrates why a pass/fail assessment is insufficient on its own for an

engineering evaluation: a BoM can pass every qualitative check while still reconstructing fewer than six in ten of its true parent-child relationships correctly, and a single root-level extraction error can propagate into a large share of the hierarchy's failures.

BoM 3 presents the strongest hierarchy result in the evaluation: Precision, Recall, F1, and Level Accuracy all reach 100%. This is a notable result given that BoM 3 was specifically selected to test complex, deeply nested hierarchies expressed through dot-notation level indicators (up to nine levels) rather than a simple numeric Level field, the dataset characteristic identified in Table 5.1 as testing hierarchy reconstruction specifically. This confirms that both the parent-child relationships and the depth of every component within the hierarchy were reconstructed without error, despite the added complexity of BoM 3's non-standard, dot-notation level format.

BoM 6 illustrates a different failure mode entirely, hallucinated rather than missing structure. BoM 6's source BoM is explicitly flat, with no hierarchy indicator present. The generated output nonetheless introduced 120 parent-child relationships not present in the source document, nesting 71 top-level components beneath invented "children" arrays. Because the reference contains no true relationships against which to measure recall, Precision is defined and reported as 0.000, while Recall and F1 are mathematically undefined rather than zero. This result is worth distinguishing from BoM 5's: where BoM 5 under-produces the correct hierarchy as scale increases, BoM 6 over-produces hierarchy where none should exist, suggesting the hierarchy rules embedded in the extraction prompt are applied by the model even when no Level attribute is present to license them.

BoM 4 expresses its hierarchy through the keyword "assieme" rather than a numeric Level field, and the hierarchy and Level Accuracy checks, which depend on a numeric Level column, could not be applied. This is a limitation

of the validation procedure's applicability to this BoM's format, not a pipeline failure.

Taken together, the six BoMs show that extraction accuracy is consistently high, with BoM 5's scale as the sole significant exception, while hierarchy accuracy is considerably more variable, ranging from hallucinated structure (BoM 6) through under-identification at scale (BoM 5) to near-perfect reconstruction on a deeply nested, non-numeric hierarchy format (BoM 3). Notably, across every BoM where Level Accuracy could be computed, the metric remained at or near 100%, indicating that once a parent-child relationship is correctly identified by the pipeline, it is reliably positioned at the correct depth within the broader hierarchy, the pipeline's principal weakness lies in relationship *identification*, not in structural placement once identification succeeds.

5.3 Industrial case study validation: Methodology Application in a Real-World Context

While Section 5.2 assessed whether the proposed methodology performs correctly across a diverse set of BoMs, this section has a different purpose: rather than testing correctness again, it examines where the methodology stands within a real, industrial LCI development project. Specifically, the case study illustrates three practical contributions within the conventional workflow described in Chapter 3: it accelerates and simplifies the preparation of information present in the BoM, it supports the identification of missing or incomplete information by making the product's assembly structure explicit, and it provides the structural foundation required for future automation of foreground LCI generation. The remainder of this section details how these contributions manifest for the selected product.

5.3.1 Description of the industrial case study

The selected product for the industrial case study belongs to the household appliance sector, specifically a combination oven–microwave, chosen for its structural complexity, material diversity, and the challenges it presents for foreground life cycle inventory (LCI) development.

Household appliances are a representative category for this case study for three reasons. First, they combine mechanical and electronic/electrical components within a single product, resulting in a structurally and functionally complex system rather than a simple mechanical assembly. Second, they are assembled from components manufactured through a wide range of production processes and materials, including plastics, metals, glass, and electronic components, reflecting the diversity of manufacturing options encountered in industrial practice more broadly. Third, their bills of materials typically combine components manufactured internally with components purchased from external suppliers, a distinction that is directly relevant for foreground inventory development, since internally manufactured and externally sourced components generally require different approaches to material and process identification.

Within this category, the combination oven–microwave was selected specifically because it integrates two distinct cooking technologies, conventional thermal convection and microwave radiation, within a single product. This hybrid configuration adds a further layer of complexity beyond that already characteristic of household appliances in general, combining components specific to each cooking technology alongside shared structural and electronic elements, and making the resulting BoM a demanding case for automated data extraction and structuring.

Household appliances are also well suited to this kind of validation because a reference foreground inventory can be developed transparently and

reproducibly: relevant background inventory data are available in established environmental databases, and primary data, such as component weights and material composition, can be obtained directly through manual product disassembly, technical documentation analysis, and manufacturer specifications. This availability of transparent, reproducible reference information was a key criterion in selecting this product, since it provides a reliable benchmark against which the output of the proposed methodology could be compared.

5.3.2 Reference foreground inventory: preparation and manual disassembly

To assess the applicability of the proposed AI-assisted methodology, a reference foreground inventory was developed for the selected combination oven–microwave through the conventional approach described in general terms in Chapter 3. This section describes how that reference was obtained for this specific product, with particular attention to the preparation phase, the analysis and structuring of the BoM that precedes physical disassembly, since this is the specific activity, the proposed methodology is intended to support.

The industrial BoM for this product was first analyzed to identify its structure, and specifically the parent–child relationships encoded within it through the Level attribute and the ordering of entries. This analysis produced an initial understanding of the product's assemblies and subassemblies, which was then used to guide the physical dismantling of the appliance, proceeding from high-level assemblies toward individual components. Each disassembly step served a dual purpose: dismantling the physical product, and simultaneously verifying the structure previously derived from the BoM. During each step, the identified components and sub-assemblies were checked against the BoM-derived structure, confirming component presence, parent–child

relationships, and assembly levels. This iterative cross-checking between the BoM and the physical product, consistent with the general workflow described in Chapter 3, established a verified correspondence between the digital and physical product representations, which constitutes the reference structure used in Section 5.3.3.

Once this structure was established and verified, material composition, manufacturing processes, and component mass were identified for each component, following the general procedure described in Chapter 3. This complete disassembly resulted in the identification of seven product modules: Packaging, Body, Door, and Control Panel as external assemblies, and Cooling, Cavity, and Electronic Components as internal assemblies. The Packaging, Body, and Door modules are illustrated in Figures 5.4–5.6 as representative examples of different levels of structural complexity; full component-level data for all seven modules, including material, manufacturing process, mass, and quantity for each component, is provided in **Appendix A**. This verified structure and its component-level data constitute the reference used for the automated-output comparison presented in **Section 5.3.4**.



Figure 5.4 Example of the packaging module and the identified components considered during manual inventory development.

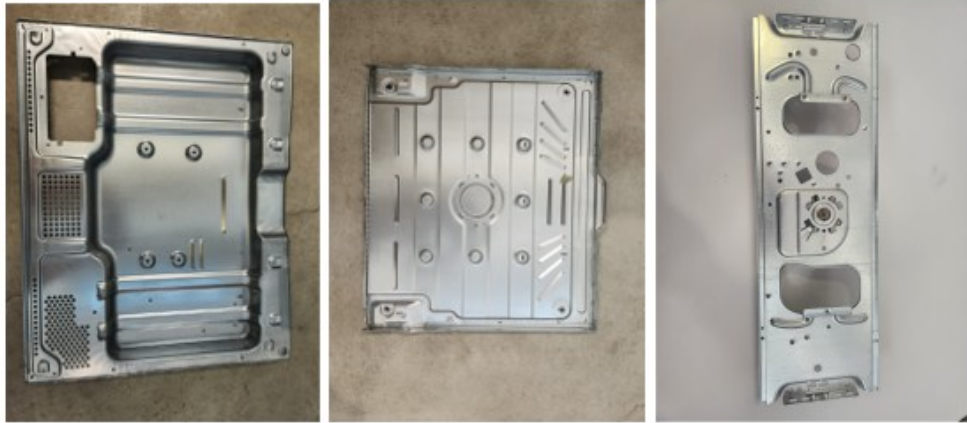


Figure 5.5 Examples of the metallic components of the Body module, including zinc-coated panels.



Figure 5.6 Door module disassembly and the main components identified during manual inventory development.

5.3.3 Application of the proposed methodology to the case-study BoM

The industrial BoM used in this case study was provided in its original manufacturing-oriented format. The document contained information related to product components, including article identifiers, descriptions, quantities, hierarchical levels, and additional engineering attributes. As typically observed in industrial documentation, the product structure was represented through tabular information, where component relationships were implicitly defined through level indicators and the ordering of entries rather than being

made explicit. Figure 5.7 shows a portion of the BoM in its original presentation, illustrating this implicit organization.

	A	B	C	D	E	F	G	H
1	DTEN - 20240528							
2	Articolo	944005046	Tp.(BC)	22				
3	PARTDESC	OKO7N401B						
4	DEF							
5	Data Validita'	2024/05/28						
6								
7	#	Livello	Articolo	Grup.	Seq.	Descrizione	Qta'	
8	1	0	944005046			OKO7N401B	1	
9	2	1	400504600	000	0000	ELC ML 00	1	
10	3	1	A23059114	010	0000	BODY MODUL MW TT EU	1	
11	4	2	A20186502	010	0010	TRAVERSE REAR MW	1	
12	5	3	555122402	000	0000	COIL0,8X680DX53D Z100 MBC L212	1	
13	6	2	A02579401	010	0020	SCREW 4,2X16 SS A2	3	
14	7	2	A23619401	010	0030	PANEL SIDE PRE ASSEMBLY RIGHT	1	
15	8	3	A19512501	000	0010	PANEL SIDE RIGHT	1	
16	9	4	555122404	000	0010	COIL0,6X500DX52D Z100 MBC L580	1,4	
17	10	3	A20581103	000	0020	TRIM LATERAL RIGHT PAINTED	1	
18	11	4	A20581101	000	0010	TRIM LATERAL RIGHT RAW	1	
19	12	2	387854510	010	0040	SCREW 4,2X9,5 DIN 7981	10	
20	13	2	A23619402	010	0050	PANEL SIDE PRE ASSEMBLY LEFT	1	
21	14	3	A19512502	000	0010	PANEL SIDE LEFT	1	

Figure 5.7 Portion of the original industrial BoM, illustrating the implicit representation of hierarchical relationships through level indicators and row ordering.

The BoM was processed following the methodology described in Chapter 4: identification of the relevant columns describing component names, quantities, hierarchy levels, and other engineering attributes; extraction of the corresponding component information independently of the original spreadsheet layout; reconstruction of the parent–child relationships between modules, assemblies, and individual components; and generation of the resulting structured JSON representation.

The extracted information was transformed into a machine-readable JSON representation containing the identified product elements and their corresponding relationships. The generated representation preserved the main attributes extracted from the original BoM, reflecting the actual field naming convention used in this BoM's engineering documentation: component identifiers (Articolo), component descriptions (Descrizione), hierarchical levels (Livello), and parent–child relationships between assemblies and

components, expressed through nested elements. The transformation from the original tabular representation into a hierarchical structure allowed the product architecture to be explicitly represented: a component identified in the BoM as belonging to a higher-level assembly was automatically organized as a child element of the corresponding parent component in the generated JSON structure. Beyond representing individual parent–child relationships, the generated hierarchy provides a global view of the product structure, enabling the identification of the different assembly levels, the number of modules composing the product, and the distribution of components within each assembly.

An example of the generated output is presented below:

```
json
{
  "Livello": 1,
  "Articolo": "A23059114",
  "Descrizione": "BODY MODUL MW TT EU",
  "children": [
    {
      "Livello": 2,
      "Articolo": "A20186502",
      "Descrizione": "TRAVERSE REAR MW"
    }
  ]
}
```

In this representation, the assembly "BODY MODUL MW TT EU" is identified as a higher-level product element, while "TRAVERSE REAR MW" is represented as a subordinate component.

5.3.4 Comparison between automated and manual results

To assess whether the structured representation generated by the proposed methodology corresponds to the physical product organization established during manual disassembly, the automated JSON output was compared component by component against the reference inventory for the Body, Door, and Packaging modules.

The industrial BoM defines the Body module ("BODY MODUL MW TT EU") as a Level-1 assembly composed of 12 subordinate components: the rear traverse, two side-panel pre-assemblies (each further decomposed into a raw panel and a painted lateral trim, i.e. Level-2 assemblies with their own Level-3 children), the bottom panel, the terminal bracket, the rear panel, the top panel, and the three-pole socket. The Door module ("DOOR MW GN MOD") is a substantially deeper structure, comprising 32 components across five hierarchy levels, including the front glass assembly, its adhesive/glue sub-processes, the inside glass glue assembly, hinges, and interlock assemblies. The Packaging module ("PACKAGING GN") comprises 9 components across two levels. Each of these structures was independently confirmed during the physical disassembly described in Section 5.3.2, where every element was identified, weighed, and matched to its corresponding BoM entry.

The automated pipeline output reproduced all three structures in full. All 12 Body components, all 32 Door components, and all 9 Packaging components were present in the generated JSON, correctly nested under their respective parent nodes; no components were missing, duplicated, or misassigned to an incorrect parent in any of the three modules, including the Door module's deeper, five-level sub-hierarchy (e.g., the individual adhesive components nested under "DOOR GLUE PROCESS GN" and "DOOR INNER CO GLUE PROCESS GN"). Table 5.4 summarizes this result.

Table 5.4 Component-level comparison between the automated pipeline output and the manually verified Body, Door, and Packaging modules

Module	Components in source BoM	Components in automated output	Hierarchy correctly preserved	Result
Body	12	12	Yes (all levels)	Full match
Door	32	32	Yes (all levels)	Full match
Packaging	9	9	Yes (all levels)	Full match

This result provides direct, component-level evidence, rather than a general impression, that across three modules of varying size and structural depth, including a five-level hierarchy in the Door module, the automated output is a faithful digital counterpart of the physically verified product structure, and could be used as-is as the structural basis for assigning material and process information during foreground LCI preparation. A corresponding component-level comparison for the remaining four modules (Control Panel, Cooling, Cavity, Electronic Components), all documented in Appendix A, was not carried out as part of this validation and is left for future verification.

5.3.5 From extracted structure to foreground LCI: a worked example

Having confirmed in Section 5.3.4 that the generated structure faithfully reproduces the physical product organization, this section examines a single component in detail to show concretely what that structure replaces, what it makes visible, and what remains to be added toward a complete foreground LCI.

Replacing the manual preparation step. In the original BoM, component A02579401 ("SCREW 4,2X16 SS A2") appears as a single tabular row: a Level value of 2, positioned beneath the row for "BODY MODUL MW TT EU" (Level 1), with its assembly membership indicated only by that level

number and its position in the row sequence. Determining that this screw belongs to the Body module, rather than, say, being an independent top-level item, requires a practitioner to read the level numbers and infer the nesting, exactly the interpretation step described in Section 5.3.2. In the generated output, this same relationship is made explicit without that inference step:

```
json
{
  "Livello": 2,
  "Articolo": "A02579401",
  "Descrizione": "SCREW 4,2X16 SS A2",
  "Qta": 3.0
}
```

nested directly inside the children array of "BODY MODUL MW TT EU." The parent-child relationship that previously had to be read from level numbers and row order is now a structural fact of the JSON itself.

Making missing information explicit. Comparing this generated record against the same component's entry in the reference inventory (Appendix X) shows exactly what the methodology's output does and does not contain:

Field	Generated output (Section 5.3.3)	Reference inventory (Appendix X)
Identifier	A02579401	A02579401
Description	SCREW 4,2X16 SS A2	SCREW 4,2X16 SS A2
Quantity	3.0	3
Material	—	Carbon Steel
Material (Ecoinvent)	—	Steel, low-alloyed {GLO} market for Cut-off, U
Manufacturing process	—	Turning
Mass	—	—

Identifier, description, quantity, and hierarchical position are present in both; material and process information, present in the manually developed reference, are absent from the generated output, because this information is not contained in the original BoM at all, it was obtained by the practitioner through physical inspection during disassembly, not through document interpretation. Because the generated structure preserves the same component-level granularity as the reference, this gap is identifiable precisely, component by component, rather than only as a general statement that "environmental data is missing" from the BoM as a whole. (This particular component also has no recorded mass in the reference inventory itself, component-level weighing was not performed for every fastener, illustrating that some gaps originate in the reference data collection, not only in the source BoM.)

Toward automated LCI generation. Because the two records share the same identifier, the missing fields could in principle be attached directly to the structure the methodology already produces:

```
json
{
  "Livello": 2,
  "Articolo": "A02579401",
  "Descrizione": "SCREW 4,2X16 SS A2",
  "Qta": 3.0,
  "Material": "Carbon Steel",
  "Manufacturing process": "Turning"
}
```

This is not something the current methodology performs automatically, the material and process values above come from the manually developed reference inventory, not from any matching step in the pipeline, but it illustrates concretely what the generated structure is a foundation *for*: each

component in the output is already uniquely identified and correctly positioned in the product hierarchy, which is the precondition for attaching material, process, and eventually background-database information to it. The methodology therefore represents a first step toward automated foreground LCI generation, providing the structural scaffold onto which the remaining information, obtained today through manual inspection, and discussed as a direction for future automation in Chapter 6, would need to be attached.

6. DISCUSSION, CONCLUSION AND FUTURE PERSPECTIVES

6.1 Overall Discussion

The increasing availability of digital engineering data within modern manufacturing environments presents significant opportunities for improving the efficiency of LCA. BoMs, generated and maintained throughout the product development process, contain detailed information describing product composition, quantities, and hierarchical relationships. Despite this abundance of engineering information, the development of foreground LCI data remains a predominantly manual activity that requires extensive interpretation of engineering documentation before environmental information can be assigned. As established in Chapter 3, this interpretation step, analyzing the BoM's implicit structure before physical disassembly and material identification can proceed, is itself a recurring source of manual effort, particularly given the inconsistency of hierarchy presentation observed across different BoMs.

To address this, this research proposed an AI-driven methodology that transforms heterogeneous industrial BoMs into standardized, hierarchical, machine-readable representations, focusing specifically on this preparatory interpretation step rather than on the complete LCI generation process.

The technical validation presented in Chapter 5 shows that this focus is justified but not uniformly straightforward. Extraction accuracy was consistently strong across the evaluated BoMs, with F1-scores at or above 95.66% for five of the six cases. Hierarchy reconstruction proved considerably more variable, and the specific pattern of that variability is itself an informative result. BoM 3, deliberately selected to test a complex, dot-notation hierarchy format rather than a simple numeric Level field, achieved a perfect result, 100% Precision, Recall, F1, and Level Accuracy,

demonstrating that the methodology's hierarchy-reconstruction logic is not inherently limited by non-standard level encodings. BoM 5, the largest and most deeply nested case (360 components, six hierarchy levels), showed markedly lower hierarchy Recall (23.06%), but its Level Accuracy remained 100% among the relationships that were correctly captured, indicating that the pipeline's difficulty at this scale lies specifically in identifying relationships, not in misplacing components once a relationship is found. BoM 6, whose source document has no hierarchy indicator at all, revealed a different failure mode entirely: the pipeline introduced 120 parent-child relationships that do not exist in the source, hallucinating structure rather than failing to find it. BoM 1's result illustrates why this quantitative layer matters at all: it was recorded as a full qualitative success, yet its hierarchy F1 (57.96%) was traced to a single root-level extraction error that cascaded into 51 of its 52 hierarchy failures, a finding a stage-level pass/fail check could not have surfaced.

Within the industrial case study, the automated output reproduced the physically verified structure of the Body, Door, and Packaging modules in full, 53 components across three modules of varying size and structural depth, including the Door module's five-level sub-hierarchy, providing direct, component-level evidence that the generated representation can serve as a reliable structural basis for a real product, not only for the synthetic and industrial BoMs used in the technical validation. Taken together, these findings indicate that Artificial Intelligence can effectively support the transformation of industrial engineering documentation into structured, LCI-oriented information, within a scope and set of limits that this validation makes explicit rather than assumes: reliable extraction across varied formats, hierarchy reconstruction that is strong except at scale or in the complete absence of a hierarchy signal, and no current mechanism for catching the kind of malformed output that caused BoM 1's cascade. Expert knowledge remains

essential for material verification, environmental dataset selection, and inventory validation, but the results demonstrate that a substantial and previously manual portion of the preparation workflow can be automated for BoMs within the tested range of size and structure.

The following sections address the research questions directly, summarize the scientific and practical contributions, identify the principal limitations, and outline directions for future research.

6.2 Answers to Research Questions

RQ1. How have Artificial Intelligence techniques been applied to Life Cycle Assessment and Life Cycle Inventory generation, and what research gaps remain in the automation of foreground Life Cycle Inventory development?

The literature review (Chapter 2) showed that AI has been applied across several LCA-related tasks, impact prediction, missing-data estimation, and, more recently, LLM-based structured extraction across visually rich documents, heterogeneous tables, and unstructured text. Within the LCI domain specifically, existing AI applications either produce deliberately flat, atomic output (Bejjit et al., 2026) or assume a structured foreground representation already exists as their starting point (Castle et al., 2025), (Bertrand et al., 2025a). None of the reviewed approaches reconstruct hierarchical product structure from a structured but non-standardized industrial BoM using an explicit structural signal, the Level attribute, the way this thesis does, because the document types those approaches target (invoices, general tables, scientific text) have no equivalent signal to exploit. This gap, identified from the literature and independently reinforced by the research team’s own industrial experience documented in Chapter 3, motivated the methodology developed in Chapter 4.

RQ2. How can an AI-driven methodology automatically transform heterogeneous industrial Bills of Materials into structured representations suitable for foreground Life Cycle Inventory development?

The proposed methodology combines deterministic preprocessing automated BoM and header detection, metadata separation, dynamic schema generation with an LLM-based extraction and hierarchical reconstruction stage that interprets the BoM's Level attribute according to an explicit parent-assignment rule. The technical validation confirmed this design's adaptability directly: the six evaluated BoMs spanned English and Italian field-naming conventions, hierarchy encodings ranging from explicit numeric levels to dot-notation depth markers (BoM 3) to keyword-based inference (BoM 4), and layouts ranging from standard top-of-sheet tables to offset positions with preceding metadata (BoM 2), and the methodology adapted to all of these without per-format configuration. At the same time, the validation clarified where this adaptability breaks down: hierarchy reconstruction failed differently depending on the cause, under-identification as document size and depth increased (BoM 5), and hallucinated relationships when no hierarchy signal was present at all (BoM 6), indicating that the boundary condition for this methodology is not simply "large BoMs" but specifically the presence, absence, and scale of the structural signal the methodology relies on.

RQ3. To what extent can the proposed methodology support foreground Life Cycle Inventory development in industrial practice?

The technical validation demonstrated reliable extraction across the tested BoMs and hierarchy reconstruction that was strong in four of six cases, with the two exceptions each informative in a different way under-identification at scale (BoM 5) and hallucination in the absence of a hierarchy signal (BoM 6). Within the industrial case study, the automated output was compared component by component against the physically verified Body, Door, and

Packaging modules of the reference product, showing complete correspondence between the generated structure and the manually established inventory across all three, including the Door module's five-level hierarchy; the corresponding comparison for the remaining four modules (Control Panel, Cooling, Cavity, Electronic Components) was not carried out and remains an open point for future verification. The generated representation reduced the manual interpretation effort associated with understanding each BoM's structure by making explicit what the source document only encodes implicitly, and as BoM 1's traced root-cause failure shows, the structuring process itself surfaces specific extraction errors rather than obscuring them behind an aggregate pass/fail label. Although the methodology does not automate environmental dataset selection or complete LCI generation, and its validation does not yet extend to processing-time or run-to-run consistency benchmarking, the results demonstrate that it can reliably support the preparation stage of industrial LCI development for BoMs within the tested range of size and structural complexity.

Overall, the three research questions are answered within a validated scope rather than a general claim: a specific, identified research gap (RQ1), an original methodology whose adaptability and boundary conditions are both demonstrated with evidence rather than asserted (RQ2), and a concrete, component-level demonstration of practical value alongside an honest account of where that value currently ends (RQ3).

6.3 Scientific and Practical Contributions

6.3.1 Scientific Contributions

The first contribution of this research is the identification, through a systematic review of the literature (Chapter 2), of a specific and previously unaddressed gap: while AI has been applied to environmental impact

prediction, missing-data estimation, semantic analysis, and background-database matching within LCA, the automated interpretation and hierarchical restructuring of industrial engineering documentation, specifically the exploitation of a pre-existing structural signal such as a BoM's Level attribute had not been addressed by either the document-understanding literature or the existing LCI-domain AI applications reviewed. This clarifies where, within the broader landscape of AI-assisted LCA research, the preparation stage of foreground inventory development sits, and establishes it as a distinct research direction rather than a special case of general document extraction.

6.3.2 Methodological Contributions

The principal contribution of this thesis is the AI-driven methodology itself: a two-stage architecture combining deterministic preprocessing with LLM-based extraction and hierarchical reconstruction, designed specifically to exploit a BoM's Level attribute as an explicit structural signal rather than treating hierarchy as something to be inferred from unstructured text.

A second, distinct methodological contribution lies in the validation design developed in Chapter 5. Rather than relying on a single pass/fail judgment per BoM, the validation separates entity-level and attribute-level correctness from hierarchy-level correctness, using Precision, Recall, and F1-score computed from True Positive, False Positive, and False Negative counts at each level, together with a secondary Level Accuracy check that isolates depth-consistency from relational correctness. This separation is what made it possible to distinguish BoM 1's near-perfect extraction from its weak hierarchy score, and to distinguish BoM 5's under-identification from BoM 6's hallucination, two failure modes that a single aggregate accuracy figure would have reported identically. This three-level, TP/FP/FN-based evaluation framework, combined with the qualitative stage-level assessment it

complements, constitutes a reusable evaluation approach for future work on AI-assisted BoM interpretation, independent of the specific methodology it was used to validate here.

6.3.3 Practical Implications

From an industrial perspective, the proposed methodology demonstrates that AI can support the interpretation of complex engineering documentation without requiring organizations to modify their existing BoM formats or engineering information systems. The six technical-validation BoMs and the industrial case-study BoM were all processed in their original, unmodified layouts.

The validation results summarized in Section 6.1 translate into a practical deployment guideline: extraction output can be trusted with limited review across the range of formats tested, while hierarchy output should receive closer scrutiny specifically for very large or deeply nested BoMs and for BoMs lacking an explicit hierarchy indicator, since these are the two conditions under which the validation identified systematic rather than isolated errors. Within that scope, the methodology provides a practical foundation for reducing the manual preparation effort that currently precedes expert-driven material verification, dataset selection, and inventory validation.

6.4 Limitations of the Proposed Methodology

While the proposed methodology demonstrates the potential of Artificial Intelligence to support foreground LCI development, its scope is intentionally limited to the transformation of heterogeneous industrial BoMs into structured, hierarchical, machine-readable representations. Several aspects of the overall LCI development process therefore remain outside the scope of

this research, and the validation itself surfaced specific, evidence-based limitations rather than only general ones.

First, the methodology does not automate the complete generation of a foreground LCI. The generated representation provides a structural foundation, but the assignment of materials, manufacturing processes, and elementary flows still requires additional data sources and expert intervention.

Second, the methodology does not perform automatic selection or validation of background datasets from databases such as Ecoinvent, and this is a more involved gap than a missing feature. As discussed in Section 5.3, linking a single component to appropriate background data typically requires several distinct datasets rather than one, for raw material production, semi-finished processing, and the manufacturing or forming process applied to the component, each of which must additionally be specified by system model (Cut-off, APOS, or Consequential) and by geographic scope, and for which no matching activity may exist at all, requiring a custom or proxy dataset. None of this dataset-selection complexity is addressed by the current methodology, nor is the prerequisite of programmatic access to the background database itself.

Third, the reliability of hierarchy reconstruction is affected by the size and structural depth of the input BoM, and by the presence or absence of a genuine hierarchy signal. The technical validation demonstrated two distinct manifestations of this: BoM 5 (360 components, six hierarchy levels) showed markedly reduced hierarchy Recall, consistent with the current single-pass implementation, which processes each BoM within a single model call without a chunking mechanism; BoM 6, which has no hierarchy indicator at all, showed the methodology introducing hierarchical structure that does not

exist in the source. Both are concrete, evidence-based boundary conditions on the methodology's current reliability rather than assumed limitations.

Fourth, the current implementation does not apply automated parsing or schema validation to the LLM's output. This is not a hypothetical concern: BoM 1's hierarchy result an F1-score of 57.96% despite a near-perfect extraction score was traced directly to a single root-level component whose identifier was generated in a malformed form, which in turn caused 51 of its 52 hierarchy failures. Because no automated validation step exists to catch this class of error, it surfaced only through the detailed quantitative comparison performed in this validation, and would not be caught automatically in a production deployment without such a mechanism. Introducing an automated validation-and-repair loop, for example, JSON schema validation with retry-on-failure, is identified as a direction for future work in Section 6.5.

Fifth, the methodology's output quality depends on the quality and completeness of the input BoM. The methodology preserves and structures the information available in the source document but cannot infer engineering information that is missing or incorrect in that source.

Sixth, the quantitative validation conducted in this work establishes entity, attribute, and hierarchy accuracy, but does not extend to processing time, scalability under batch conditions, or consistency across repeated runs on the same input, three of the five criteria in the benchmarking framework proposed by (Bertrand et al., 2025b). Establishing these remaining dimensions would allow a more direct, quantitative comparison against manual and alternative automated approaches.

Seventh, the industrial case study's component-level comparison was completed for the Body, Door, and Packaging modules; a corresponding comparison for the remaining four modules (Control Panel, Cooling, Cavity,

Electronic Components), although all were characterized during the manual disassembly and are documented in Appendix A, was not carried out. The industrial case study is also limited to a single representative product. The conclusions drawn from it are therefore best understood as demonstrated for three of seven modules of one product rather than validated across the full product or across a product category.

Finally, despite these capabilities, expert knowledge remains essential throughout foreground LCI development for material verification, interpretation of ambiguous engineering information, environmental dataset selection, and final inventory validation. The proposed methodology should be regarded as a decision-support tool that reduces manual preprocessing effort within a demonstrated and bounded scope, not as a replacement for that expertise.

6.5 Future Research Directions

The proposed methodology represents an initial step toward automating foreground LCI generation by improving the accessibility and structuring of industrial product information. Several research directions follow directly from the specific limitations identified in Section 6.4, rather than from general aspiration.

The most immediate, evidence-backed priority is addressing the scalability limitation demonstrated by BoM 5: since the current pipeline processes each BoM in a single pass without a chunking mechanism, future work should investigate chunking or hierarchical decomposition strategies capable of processing large and deeply nested BoMs, such as the 360-component case evaluated in this work, without the under-identification observed at that scale.

A second, equally direct priority is introducing automated output validation. BoM 1's traced failure, a single malformed identifier cascading into the

majority of that BoM's hierarchy errors, demonstrates concretely why this matters: a JSON schema validation and retry-on-failure mechanism could catch this class of error before it reaches the practitioner, rather than requiring, as in this validation, a detailed quantitative comparison to surface it after the fact.

A third priority concerns extending the quantitative benchmarking established in this work along the dimensions it does not yet cover — processing time, scalability under batch conditions, and consistency across repeated runs, completing the five-criterion framework proposed by (Bertrand et al., 2025a) and enabling direct, quantitative comparison against manual and alternative automated approaches.

A further direction concerns extending the methodology beyond structural reconstruction toward a more complete automated LCI workflow, integrating AI-based modules capable of identifying materials, manufacturing processes, and relevant environmental exchanges from the reconstructed product information. As Section 6.4 makes clear, this is a substantially more involved problem than adding a single background-dataset reference per component: it requires resolving, for each component, which of several linked datasets apply, under which system model and geographic scope, and how to handle components for which no matching dataset exists. Future research could investigate hybrid approaches combining LLM-based semantic understanding with domain-specific knowledge bases and existing databases such as Ecoinvent to address this multi-dataset selection problem directly, while maintaining the transparency and traceability that dataset-selection decisions require.

Related to this, the management of uncertainty represents an important direction in its own right. Automated LCI generation inherently involves incomplete information and ambiguous component descriptions — the kind

of ambiguity discussed in Section 6.4 for material and dataset assignment. Future work should incorporate mechanisms for identifying missing information, quantifying confidence, and prioritizing human review, supporting a human-in-the-loop workflow in which the automated methodology accelerates preparation while experts remain involved in validation.

Further research should also extend the industrial case study format used in this thesis: completing the component-level comparison for the remaining four modules (Control Panel, Cooling, Cavity, Electronic Components) left open in Section 5.3, and extending the validation to additional products and manufacturing sectors, to assess robustness under a broader range of data formats and organizational practices than the six BoMs and single product evaluated here.

Finally, integrating the proposed methodology within industrial PLM and digital engineering environments represents a promising direction, since BoMs are continuously generated and maintained throughout product development. Connecting AI-assisted LCI preparation directly to existing engineering data infrastructure could support earlier, more continuous environmental assessment within the design process itself.

Overall, these directions point toward a more complete AI-supported LCI generation framework, one in which structural extraction, environmental dataset selection, uncertainty management, and inventory validation are integrated into a coherent workflow, building directly on the structural foundation, and the validated boundary conditions, established in this thesis.

BIBLIOGRAPHY

- Babbitt, C. W., Madaka, H., Althaf, S., Kasulaitis, B., & Ryen, E. G. (2020). Disassembly-based bill of materials data for consumer electronic products. *Scientific Data* 2020 7:1, 7(1), 251-. <https://doi.org/10.1038/s41597-020-0573-9>
- Bai, F., Kang, J., Stanovsky, G., Freitag, D., Dredze, M., & Ritter, A. (2024). Schema-Driven Information Extraction from Heterogeneous Tables. *EMNLP 2024 - 2024 Conference on Empirical Methods in Natural Language Processing, Findings of EMNLP 2024*, 10252–10273. <https://doi.org/10.18653/V1/2024.FINDINGS-EMNLP.600>
- Bejjit, C., Hennioui, D., Lai, F., Kroiss, A., Mas-Fons, A., Guth, B., Muller, S., Climent, D. M., Lutter, S., Maus, V., & Beylot, A. (2026). *Harnessing the potential of AI for LCI data compilation from company reporting*. <https://doi.org/10.21203/rs.3.rs-9280837/v1>
- Bertrand, A., Bauer, T., Charbuillet, C., Bonte, M., Voyer, M., & Perry, N. (2025a). Automation in LCA, reconciliation of inventory data: Case study on manufactured products subject to the French RE2020 building regulation to validate the proposed method. *Procedia CIRP*, 135, 69–75. <https://doi.org/10.1016/j.procir.2024.12.003>
- Brahem, S., & Favi, C. (2025). LCA comparison of freestanding kitchens adopting three technologies: electric, gas and induction. *Procedia CIRP*, 135, 101–105. <https://doi.org/10.1016/J.PROCIR.2024.12.010>
- Brahem, S., Rodríguez, N. B., & Favi, C. (2024). LCA analysis of a freestanding cooker: environmental assessment and comparison with other cooking appliances. *Procedia CIRP*, 122, 819–824. <https://doi.org/10.1016/J.PROCIR.2024.02.025>
- Castle, S., Schneider, J. M., Hennig, L., & Rehm, G. (2025). *Entity Linking using LLMs for Automated Product Carbon Footprint Estimation*. <http://arxiv.org/abs/2502.07418>
- Dagdelen, J., Dunn, A., Lee, S., Walker, N., Rosen, A. S., Ceder, G., Persson, K. A., & Jain, A. (2024). Structured information extraction from scientific text with large language models. *Nature Communications* 2024 15:1, 15(1), 1418-. <https://doi.org/10.1038/s41467-024-45563-x>
- Damla Turhan, G. (n.d.). *Life Cycle Assessment for the Unconventional Construction Materials in Collaboration with a Large Language Model*.
- Diamantini, C., Potena, D., Rossetti, C., & Storti, E. (2025a). *A Knowledge Graph Framework for Impact Calculation in Life-Cycle Assessment*. https://green-business.ec.europa.eu/environmental-footprint-methods/life-cycle-assessment-ef-methods_en

- Fang, X., Sun, W., Li, W., Ma, G., Wang, P., & Zuo, C. (2024). Process-based life cycle inventory framework for assessing the carbon footprint of products from complex production paths: Case of dual-phase automotive strip steel. *Journal of Cleaner Production*, 447. <https://doi.org/10.1016/j.jclepro.2024.141551>
- Finnveden, G., Hauschild, M. Z., Ekvall, T., Guinée, J., Heijungs, R., Hellweg, S., Koehler, A., Pennington, D., & Suh, S. (2009). Recent developments in Life Cycle Assessment. *Journal of Environmental Management*, 91(1), 1–21. <https://doi.org/10.1016/J.JENVMAN.2009.06.018>
- Forth, K., Abualdenien, J., & Borrmann, A. (2023). Calculation of embodied GHG emissions in early building design stages using BIM and NLP-based semantic model healing. *Energy and Buildings*, 284. <https://doi.org/10.1016/j.enbuild.2023.112837>
- Gachkar, S., Gachkar, D., Ghofrani, E., García Martínez, A., & Angulo Bahón, C. (2025). Text-based algorithms for automating life cycle inventory analysis in building sector life cycle assessment studies. *Journal of Cleaner Production*, 486. <https://doi.org/10.1016/j.jclepro.2024.144448>
- Ghoroghi, A., Rezgui, Y., Petri, I., & Beach, T. (2022). Advances in application of machine learning to life cycle assessment: a literature review. *International Journal of Life Cycle Assessment*, 27(3), 433–456. <https://doi.org/10.1007/s11367-022-02030-3>
- Giglou, H. B., D’Souza, J., Engel, F., & Auer, S. (2024). *LLMs4OM: Matching Ontologies with Large Language Models*. <http://arxiv.org/abs/2404.10317>
- Greif, L., Hauck, S., Kimmig, A., & Ovtcharova, J. (2025). A Knowledge Graph Framework to Support Life Cycle Assessment for Sustainable Decision-Making. *Applied Sciences (Switzerland)*, 15(1). <https://doi.org/10.3390/app15010175>
- GUYON, S., LEGOUBÉ, L.-A., TERRIER, P., & RIVEST, L. (2024). *The Role of CAD and PLM in Ecodesign: A Short Review*. 43–77. https://doi.org/10.1007/978-3-031-53521-5_4
- International Reference Life Cycle Data System (ILCD) Handbook - Publications Office of the EU*. (n.d.). Retrieved August 11, 2026, from <https://op.europa.eu/en/publication-detail/-/publication/325e9630-8447-4b96-b668-5291d913898e/language-en>
- ISO 14040:2006 - Environmental management — Life cycle assessment — Principles and framework*. (n.d.). Retrieved August 10, 2026, from https://www.iso.org/standard/37456.html?utm_source=chatgpt.com
- Kaushal, I., & Chakrabarti, A. (2023). System Modelling for Collecting Life Cycle Inventory (LCI) Data in MSMEs Using a Conceptual Model for Smart Manufacturing Systems (SMSs). *International Journal of*

- Precision Engineering and Manufacturing - Green Technology*, 10(3), 819–834. <https://doi.org/10.1007/s40684-022-00489-x>
- Kiemel, S., Rietdorf, C., Schutzbach, M., & Miehe, R. (2022). How to Simplify Life Cycle Assessment for Industrial Applications—A Comprehensive Review. In *Sustainability (Switzerland)* (Vol. 14, Number 23). MDPI. <https://doi.org/10.3390/su142315704>
- Kiritsis, D. (2011). Closed-loop PLM for intelligent products in the era of the Internet of things. *CAD Computer Aided Design*, 43(5), 479–501. <https://doi.org/10.1016/j.cad.2010.03.002>
- Köck, B., Friedl, A., Serna Loaiza, S., Wukovits, W., & Mihalyi-Schneider, B. (2023). Automation of Life Cycle Assessment—A Critical Review of Developments in the Field of Life Cycle Inventory Analysis. In *Sustainability (Switzerland)* (Vol. 15, Number 6). MDPI. <https://doi.org/10.3390/su15065531>
- Kuczenski, B., Mutel, C., Srocka, M., Scanlon, K., & Ingwersen, W. (2021). Prototypes for automating product system model assembly. *The International Journal of Life Cycle Assessment* 2021 26:3, 26(3), 483–496. <https://doi.org/10.1007/S11367-021-01870-9>
- Liu, Y., & Syberfeldt, A. (2022). Linking Simulation-Based LCA to Manufacturing Decision Support: An Iron Foundry Case Study. *Advances in Transdisciplinary Engineering*, 25, 325–333. <https://doi.org/10.3233/ATDE220611>
- Marwah, M., Shah, A., Bash, C., Patel, C., & Ramakrishnan, N. (2011). Using data mining to help design sustainable products. *Computer*, 44(8), 103–106. <https://doi.org/10.1109/MC.2011.257>
- Mba Wright, M., Tan, E. C. D., Tu, Q., Martins, A., Parvatker, A. G., Yao, Y., Sunol, A., & Smith, R. L. (2024). Life Cycle Inventory Availability: Status and Prospects for Leveraging New Technologies. In *ACS Sustainable Chemistry and Engineering* (Vol. 12, Number 34, pp. 12708–12718). American Chemical Society. <https://doi.org/10.1021/acssuschemeng.4c02519>
- Oliveira, F. B. de, Nordelöf, A., Sandén, B. A., Widerberg, A., & Tillman, A. M. (2022). Exploring automotive supplier data in life cycle assessment – Precision versus workload. *Transportation Research Part D: Transport and Environment*, 105. <https://doi.org/10.1016/j.trd.2022.103247>
- Peng, T., Gao, L., Agbozo, R. S. K., Xu, Y., Svyarenko, K., Wu, Q., Li, C., & Tang, R. (2024). Knowledge graph-based mapping and recommendation to automate life cycle assessment. *Advanced Engineering Informatics*, 62. <https://doi.org/10.1016/j.aei.2024.102752>
- Perot, V., Kang, K., Luisier, F., Su, G., Sun, X., Boppana, R. S., Wang, Z., Wang, Z., Mu, J., Zhang, H., Lee, C.-Y., & Hua, N. (2024). LMDX: Language Model-based Document Information Extraction and Localization. <http://arxiv.org/abs/2309.10952>

- Pham, D. H., Park, S., & Ahn, Y. (2024). A Natural language processing based machine learning approach on building material eco-label databases wrangling. *International Journal of Sustainable Building Technology and Urban Development*, 15(3), 367–380. <https://doi.org/10.22712/susb.20240026>
- Preuss, N., Alshehri, A. S., & You, F. (2024). Large language models for life cycle assessments: Opportunities, challenges, and risks. *Journal of Cleaner Production*, 466. <https://doi.org/10.1016/j.jclepro.2024.142824>
- Romeiko, X. X., Zhang, X., Pang, Y., Gao, F., Xu, M., Lin, S., & Babbitt, C. (2024). A review of machine learning applications in life cycle assessment studies. In *Science of the Total Environment* (Vol. 912). Elsevier B.V. <https://doi.org/10.1016/j.scitotenv.2023.168969>
- Saad, M., Zhang, Y., Tian, J., & Jia, J. (2023). A graph database for life cycle inventory using Neo4j. *Journal of Cleaner Production*, 393. <https://doi.org/10.1016/j.jclepro.2023.136344>
- Saavedra-Rubio, K., Thonemann, N., Crenna, E., Lemoine, B., Caliandro, P., & Laurent, A. (2022). Stepwise guidance for data collection in the life cycle inventory (LCI) phase: Building technology-related LCI blocks. *Journal of Cleaner Production*, 366. <https://doi.org/10.1016/j.jclepro.2022.132903>
- Salla, J. V. E., de Almeida, T. A., & Silva, D. A. L. (2025). Integrating machine learning with life cycle assessment: a comprehensive review and guide for predicting environmental impacts. *International Journal of Life Cycle Assessment*. <https://doi.org/10.1007/s11367-025-02437-8>
- Scrucca, F., Baldassarri, C., Baldinelli, G., Bonamente, E., Rinaldi, S., Rotili, A., & Barbanera, M. (2020). Uncertainty in LCA: An estimation of practitioner-related effects. *Journal of Cleaner Production*, 268. <https://doi.org/10.1016/j.jclepro.2020.122304>
- Sturm, J. H. L., Gehrke, S., & Herrmann, C. (2025). *A Systematic Approach for the Reliable and Automated Selection of Life Cycle Assessment Data Sets Exemplified by the Automotive Industry*. 393–406. https://doi.org/10.1007/978-3-658-45889-8_30
- Tjong, E. F., Sang, K., & De Meulder, F. (n.d.). *Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition*. Retrieved August 25, 2026, from <http://lcg-www.uia.ac.be/conll2003/ner/>
- Wernet, G., Bauer, C., Steubing, B., Reinhard, J., Moreno-Ruiz, E., & Weidema, B. (2016). The ecoinvent database version 3 (part I): overview and methodology. *International Journal of Life Cycle Assessment*, 21(9), 1218–1230. <https://doi.org/10.1007/s11367-016-1087-8>

- Zhang, Z., Metzger, A., Mei, Y., Hähnlein, F., Englhardt, Z., Cheng, T., Abowd, G. D., Patel, S., Schulz, A., & Iyer, V. (2025). *Towards Autonomous Sustainability Assessment via Multimodal AI Agents*.
- Zhou, Z., & Tao, J. (2021). CAX/PLM/ERP-LCA integration to support product life cycle engineering: A conceptual framework. *Journal of Physics: Conference Series*, 1983(1). <https://doi.org/10.1088/1742-6596/1983/1/012008>

Appendix A — Full Component-Level Reference Inventory

This appendix presents the complete component-level reference inventory obtained through manual disassembly and analysis of the combination oven–microwave used in the industrial case study (Section 5.3), for each of the seven identified product modules, plus two standalone top-level items that do not belong to any of the seven named modules. The BoM level (Lvl) column indicates each component's depth within the product hierarchy, consistent with the Level attribute used throughout the proposed methodology. Cells left blank indicate that the corresponding information was not recorded in the reference inventory for that component. Material and manufacturing process fields are reported both as originally recorded and, where available, as mapped to Ecoinvent nomenclature.

Body

Component	Code	Lvl	Qty	Material	Material (Ecoinvent)	Main process	Main process (Ecoinvent)	Mass (g)	Notes
BODY MODUL MW TT EU	A23059114	1	1						
TRAVERSE REAR MW	A20186502	2	1	COIL Z100 MBC L212 (stainless steel)					
COIL0,8X680DX53D Z100 MBC L212	555122402	3	1						
SCREW 4,2X16 SS A2	A02579401	2	3	Carbon Steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning			
PANEL SIDE PRE ASSEMBLY RIGHT	A23619401	2	1						
PANEL SIDE RIGHT	A19512501	3	1	COIL Z100 MBC L580 (stainless steel)	ask gaggiotech 99% carbon steel	Cutting and stamping	Deep drawing, steel, 650 kN press, single stroke {GLO} market for Cut-off, U		

Component	Code	Lvl	Qty	Material	Material (Ecoinvent)	Main process	Main process (Ecoinvent)	Mass (g)	Notes
COIL0,6X500DX52D Z100 MBC L580	555122404	4	1.384						
TRIM LATERAL RIGHT PAINTED	A20581103	3	1	Carbon steel painted	Steel, low-alloyed, hot rolled {GLO} market for Cut-off, U	Cutting and stamping	Deep drawing, steel, 650 kN press, single stroke {GLO} market for Cut-off, U		
TRIM LATERAL RIGHT RAW	A20581101	4	1	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Cutting and stamping	Deep drawing, steel, 650 kN press, single stroke {GLO} market for Cut-off, U		
SCREW 4,2X9,5 DIN 7981	387854510	2	10	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning			
PANEL SIDE PRE ASSEMBLY LEFT	A23619402	2	1						
PANEL SIDE LEFT	A19512502	3	1	COIL Z100 MBC L580 (carbon steel)		Cutting and stamping	Deep drawing, steel, 650 kN press, single stroke {GLO} market for Cut-off, U		
COIL0,6X500DX52D Z100 MBC L580	555122404	4	1.384						
TRIM LATERAL LEFT PAINTED	A20581106	3	1	Carbon steel painted	Steel, low-alloyed, hot rolled {GLO} market for Cut-off, U	Cutting and stamping	Deep drawing, steel, 650 kN press, single stroke {GLO} market for Cut-off, U		
TRIM LATERAL LEFT RAW	A20581104	4	1	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Cutting and stamping	Deep drawing, steel, 650 kN press, single stroke {GLO} market for Cut-off, U		
SCREW 4,2X9,5 DIN 7981	387854510	2	10	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off		
PANEL BOTTOM	A19512401	2	1	COIL Z100 MBC L548 (carbon steel)		Cutting and stamping	Deep drawing, steel, 650 kN press, single stroke {GLO} market for Cut-off, U		
COIL0,6X595DX53D Z100 MBC L548	555122405	3	1.556						
SCREW 4,2X9,5 DIN 7981	387854510	2	4	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		
TERMINAL BRACKET EU MW	A27438803	2	1						

Component	Code	Lvl	Qty	Material	Material (Ecoinvent)	Main process	Main process (Ecoinvent)	Mass (g)	Notes
TERMINAL BRACKET EU MW	A27438803	2	1	steel/brass					
SCREW 4,2X9,5 DIN 7981	387854510	2	4	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		
PANEL REAR (back panel)	A19526101	2	1	COIL Z100 MBC L489 (carbon steel)				965	
COIL 0,5X605DX53D Z100 MBC L489	555122403	3	1.177						
SCREW 4,2X9,5 DIN 7981	387854510	2	5	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		
PANEL TOP	A19524401	2	1	COIL Z100 MBC L600 (carbon steel zinc coated)				1300	
COIL 0,5X600 DX52 Z100 MBC L600	555122401	3	1.413						
SCREW 4,2X9,5 DIN 7981	387854510	2	4	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		
SPACER	387295701	2	2						
SOCKET 3 POLES	A24411001	2	1	plastic					
SCREW 4,2X9,5 DIN 7981	387854510	2	2	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		

Cooling

Component	Code	Lvl	Qty	Material	Material (Ecoinvent)	Main process	Main process (Ecoinvent)	Mass (g)	Notes
COOLING MODULE COMBI COOK	A20242105	1	1						
CLINCHED ASSEMBLY CC+TC+BC	A21305801	2	1						

Component	Code	Lvl	Qty	Material	Material (Ecoinvent)	Main process	Main process (Ecoinvent)	Mass (g)	Notes
COMPONENT CARRIER	A18677101	3	1	COIL Z100 MBC L571 (stainless steel)	Steel, chromium steel 18/8, hot rolled {GLO} market for Cut-off, U	Cutting and stamping	Deep drawing, steel, 650 kN press, single stroke {GLO} market for Cut-off, U		
COIL0,8X593DX53D Z100 MBC L571	555122406	4	2.155						
TOP CHANNEL	A18677201	3	1	COIL MBC L420		Cutting and stamping	Deep drawing, steel, 650 kN press, single stroke {GLO} market for Cut-off, U		
COIL0,5X560DX54 Z100 MBC L420	555122408	4	0.907						
BOTTOM CHANNEL	A18677301	3	1	COIL Z100 MBC (stainless steel)	Steel, chromium steel 18/8, hot rolled {GLO} market for Cut-off, U	Cutting and stamping	Deep drawing, steel, 650 kN press, single stroke {GLO} market for Cut-off, U		
COIL0,5X508DX54 Z100 MBC	555122409	4	0.907						
FAN MOTOR ASSEMBLY 2 SPEED	A20244603	2	1						
MOTOR DOUBLE SPEED	807801109	3	1						
MOTOR PLATE	A20241501	3	1	COIL Z100 (stainless steel)	Steel, chromium steel 18/8, hot rolled {GLO} market for Cut-off, U	Cutting and stamping	Deep drawing, steel, 650 kN press, single stroke {GLO} market for Cut-off, U		
COIL 0,5X435 HX220 Z100	555122410	4	0.226						
SCREW M4X8	387873310	3	2	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		
IMPELLER	A23628501	3	1	metal		Casting	Casting, steel, lost-wax {RoW} casting, steel, lost-wax Cut-off, U		
IMPELLER DISC	A19085501	3	1	metal		Cutting and stamping	Deep drawing, steel, 650 kN press, single stroke {GLO} market for Cut-off, U		
BRACKET MAGNETRON	A21340101	2	1						
SCREW 4,2X9,5 DIN 7981	387854510	2	1	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer		

Component	Code	Lvl	Qty	Material	Material (Ecoinvent)	Main process	Main process (Ecoinvent)	Mass (g)	Notes
							numerical controlled {GLO} market for Cut-off, U		
BRACKET INVERTER	A23382101	2	1						
INVERTER	A20707601	2	1	electronic component					
DUCT RAMP	A23295501	2	1						
DUCT INVERTER-MAGNETRON	A23295401	2	1						
SCREW 4,2X9,5 DIN 7981	387854510	2	2	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		
SAFTY KLIXON 145° YELLOW	A01802605	2	1						
SCREW 4,2X9,5 DIN 7981	387854510	2	1						

Cavity

Component	Code	Lvl	Qty	Material	Material (Ecoinvent)	Main process	Main process (Ecoinvent)	Mass (g)	Notes
CAVITY MODUL TT GN	A25295801	1	1						
CAVITY WITH WAVEGUIDE TT	A20370401	2	1						
CAVITY TT ENAM	A18897003	3	1						
CAVITY TT DEGR	A18897002	4	1						
CAVITY RAW	A18897001	5	1						
CAVITY WRAPPER	A21713713	6	1	DC04ED (Carbon steel)	Steel, low-alloyed, hot rolled {GLO} market for Cut-off, U	Cutting and stamping	Deep drawing, steel, 650 kN press, single stroke {GLO} market for Cut-off, U		
0,6X1340X165 DC04ED	555122419	7	1.055						
CAVITY BOTTOM	A21705111	6	1	COIL DC04ED (Carbon steel)	Steel, low-alloyed, hot rolled {GLO} market for Cut-off, U	Cutting and stamping	Deep drawing, steel, 650 kN press, single stroke {GLO} market for Cut-off, U		

Component	Code	Lvl	Qty	Material	Material (Ecoinvent)	Main process	Main process (Ecoinvent)	Mass (g)	Notes
COIL 0,6X580 DC04ED (L: 560)	555122420	7	1.55						
CAVITY TOP MW	A21658613	6	1	DC04ED (carbon steel)	Steel, low-alloyed, hot rolled {GLO} market for Cut-off, U	Cutting and stamping	Deep drawing, steel, 650 kN press, single stroke {GLO} market for Cut-off, U		
COIL 0,6X625 DC04ED (L:600)	555122418	7	1.79						
CAVITY HANGERS	A22140812	6	2	HL-PSEUDO RAW MATERIAL					
HL-PSEUDO RAW MATERIAL	661000003	7	1						
CAVITY WASHER	A20280011	6	1	HL-PSEUDO RAW MATERIAL					
HL-PSEUDO RAW MATERIAL	661000003	7	1						
BRACKET SUPPORT TT	A23562001	6	1						
FRONT FRAME MW	A21285012	6	1	DC04ED (carbon steel) carbon steel painted (grey)	Steel, low-alloyed, hot rolled {GLO} market for Cut-off, U	Cutting and stamping	Deep drawing, steel, 650 kN press, single stroke {GLO} market for Cut-off, U		
COIL 0,8X686 DC04ED (L:490)	555122421	7	2.139						
ME2012D (1408544)	339493070	4	0.304						
WAVEGUIDE WITH MULTIMIRROR	A20137202	3	1						
EXHAUST DUCT	A21414101	2	1						
SHEET.MET.SCREW 4,2X10	387872810	2	2						
GRILL 230V-1650W	A08934224	2	1						
TOP HEATING ELEMENT GASKET	A21676901	2	1						
SCREW 4,2X9,5 DIN 7981	387854510	2	2	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer		

Component	Code	Lvl	Qty	Material	Material (Ecoinvent)	Main process	Main process (Ecoinvent)	Mass (g)	Notes
							numerical controlled {GLO} market for Cut-off, U		
WAVEGUIDE COVER - MICA	A18814301	2	1						
SCREW 4,2X9,5 DIN 7981	387854510	2	4	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		
TURNTABLE MOTOR	A13805508	2	1						
SPACER TT MOTOR	A25330101	2	1						
SCREW 4,2 X 25	A05038408	2	2	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		
MAGNETRON 2M261 PANASONIC	A20685801	2	1	electronic component					
SHEET METAL SCREW 4,2	A05038401	2	4	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		
INSULATION REAR	A17264203	2	1						
INSULATION 3 SIDE	A26723401	2	1						
70X30 G9 RECTANGULAR LAMP	A20163601	2	1						
LAMP FIXATION FRAME STEAM MW	A21381602	2	1						
LAMP COVER FRAME STEAM AND MW	A21382301	2	1						
LAMP GASKET STEAM AND MW	A21382401	2	1						
LAMP GLASS STEAM AND MW	A19992101	2	1					20	
RAISED COUNT.SCREW M4X12	387904612	2	4	Stainless steel		Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		
PT500 SENSOR GN	A14424405	2	1	electronic					

Component	Code	Lvl	Qty	Material	Material (Ecoinvent)	Main process	Main process (Ecoinvent)	Mass (g)	Notes
CUP NUT HEATING ELEMENT	561045004	2	5						
RING 230V-1750W-AISI 309	A08933945	2	1						
PLATE BACKING HEAT ELEMENT	A21873701	2	1						
GASKET RING HEAT	A21327201	2	1						
SH.MET.SCREW ST4,2X9,5-C-	387856310	2	1	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		
AIR INLET DUCT MW	A20208501	2	1						
SHEET.MET.SCREW 4,2X10	387872812	2	3	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		
HOT AIR MOTOR 230/240	555027100	2	1	stator and rotor					
HOTAIR SPRING BENT	561475010	2	1						
FAN WHEEL 50 GS	A23494603	2	1						
NUT M6L	561231207	2	1	Carbon steel or Brass					
FAN COVER CLOSED	A23369908	2	1						
FAN COVER MW DEGR	A23369907	3	1						
FAN COVER MW RAW	A23369901	4	1	DC04ED (Carbon steel)	Steel, low-alloyed, hot rolled {GLO} market for Cut-off, U	Cutting and stamping	Deep drawing, steel, 650 kN press, single stroke {GLO} market for Cut-off, U		
COIL 0,8X460 DC04ED (L: 270)	555122422	5	0.79						
ME2012D (1408544)	339493070	3	0.0304						
SHEET.MET.SCREW 4,2X10	387872810	2	4	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		

Electronic Components

Component	Code	Lvl	Qty	Material	Material (Ecoinvent)	Main process	Main process (Ecoinvent)	Mass (g)	Notes
ELECTRONIC ASS	555123409	1	1						
MEC1 PROG.	A26998002	2	1						
CIR. BOARD CLIP 160EHCBS	330647300	2	4	Motherboard	Printed wiring board, surface mounted, unspecified, Pb free {GLO} market for Cut-off, U - Waste Type WEEE				
CIRC.BOARD CLIP 16008109	330647301	2	1	Motherboard	Printed wiring board, surface mounted, unspecified, Pb free {GLO} market for Cut-off, U - Waste Type WEEE				
WIRING KIT COMBI MW TT	A23252807	2	1						
CIRCUIT DIAGRAM	A21576801	2	1						
NTC WIRING	555031103	2	1						
WIRING MAINHEATER CMW TT	A22716801	2	1						
SCR N6X1/4 ST AB PAN T20	349475501	2	2	Carbon steel	Steel, low-alloyed {GLO} marketfor Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		
FUSE + FUSE HOLDER	A25771602	2	1						
SIDE-KICK CABLE	561222202	2	1						
CABLE HOLDER SMALL	387963900	2	7						
CABLE STRAP	669915714	2	2						
RFI FILTER GUSTO WITH VDR	A27701003	2	1						
CIR. BOARD CLIP 160EHCBS	330647300	2	4	Motherboard	Printed wiring board, surface mounted, unspecified, Pb free {GLO} market for Cut-off, U - Waste Type WEEE				

Door

Component	Code	Lvl	Qty	Material	Material (Ecoinvent)	Main process	Main process (Ecoinvent)	Mass (g)	Notes
DOOR MW GN MOD	A26704406	1	1						
DOOR MW CPL	A26704306	2	1						
FRONT GLASS CPL GN	555123716	3	1	Glass	Flat glass, coated {RER} market for flat glass, coated Cut-off, U - Waste type Glass	Tempering	Tempering, flat glass {GLO} market for Cut-off, U		
DOOR FRONT GLASS PRN.	A27832505	4	1	Glass	Flat glass, coated {RER} market for flat glass, coated Cut-off, U - Waste type Glass	Tempering	Tempering, flat glass {GLO} market for Cut-off, U	2045	
COLUMN LEFT	A21672501	4	1						
COLUMN RIGHT	A21672502	4	1						
FRONTGL+COL.	A26480704	4	1	Glass	Flat glass, coated {RER} market for flat glass, coated Cut-off, U - Waste type Glass	Tempering	Tempering, flat glass {GLO} market for Cut-off, U		
DOOR GLUE PROCESS GN	555122801	4	1	Adhesive	N-olefins {GLO} market for Cut-off, U - Waste type landfill				
DOW CORNING 1200 PRIMER	311115420	5	0.1						
SILIC DOW CORNING EA2626	397092201	5	0.0114	Adhesive (silicon)	N-olefins {GLO} market for Cut-off, U - Waste type landfill				
SILIC DOW CORNING Q3-3636	358199056	5	0.0023	Adhesive (silicon)	N-olefins {GLO} market for Cut-off, U - Waste type landfill				
ZUKUNFT OPEN INFINITY BRUSH	A24395101	3	1						
SCREW ALT. FOR 561228544	387873144	3	2	Carbon steel (black)	Steel, low-alloyed {GLO} marketfor Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		
INSIDE GLASS GLUE CPL	555123601	3	1	Adhesive (silicon)	N-olefins {GLO} market for Cut-off, U - Waste type landfill				
DOOR INNER COVER	A27474501	4	1						
INSIDE GLASS	A21293501	4	1	Glass	Flat glass, coated {RER} market for flat glass, coated Cut-off, U - Waste type Glass				

Component	Code	Lvl	Qty	Material	Material (Ecoinvent)	Main process	Main process (Ecoinvent)	Mass (g)	Notes
INN. DOOR COVER GL.	A21725601	4	1						
DOOR INNER CO GLUE PROCESS GN	555122701	4	1	process??					
SILIC DOW CORNING EA2626	397092201	5	0.0114	Adhesive (silicon)	N-olefins {GLO} market for Cut-off, U - Waste type landfill				
SILIC DOW CORNING Q3-3636	358199056	5	0.0023	Adhesive (silicon)	N-olefins {GLO} market for Cut-off, U - Waste type landfill				
CENTER GLASS GN PRINT	A21630002	3	1	Glass	Flat glass, coated {RER} market for flat glass, coated Cut-off, U - Waste type Glass				
GLASS HOLDER CLOSURE PART	A27685801	3	1 (but in reality we have 2 of these)	metal or plastic					
CHOKE LAMELLA	A23355301	3	1						
SCREW PT 4X10 T20	349479701	3	4	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		
GASKET DOOR COVER	A27853201	3	1	Carbon Steel (black)					
SCREW PT 4X10 T20	349479701	3	2	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		
HINGE MW	A19980401	2	2					260	
SCREW M4X5 T20	A18029001	2	6	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		
INTERLOCK ASSEMBLY RH	A21243201	2	1						

Component	Code	Lvl	Qty	Material	Material (Ecoinvent)	Main process	Main process (Ecoinvent)	Mass (g)	Notes
INTERLOCK ASSEMBLY LH	A21283401	2	1						
SCREW 4,2X9,5 DIN 7981	387854510	2	2	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		
DOOR ASS MW	A21129204	2	1						

Control Panel

Component	Code	Lvl	Qty	Material	Material (Ecoinvent)	Main process	Main process (Ecoinvent)	Mass (g)	Notes
CONT.PANEL CPL. GN	A26672709	1	1					1055	
PANEL CPL.	555123510	2	1						
SUPPORTING PLATE	A23091915	3	1	stainless steel					
SUPPORTING PLATE	A23091912	4	1	plastic PBT GF10					
GLASS PANEL PRINTED	A27844305	3	1	Glass	Flat glass, coated {RER} market for flat glass, coated Cut-off, U - Waste type Glass				
LUX TIMER FRAME GLASS	A18111709	3	1	Glass	Flat glass, coated {RER} market for flat glass, coated Cut-off, U - Waste type Glass				
KNOB HOUSING	A13194505	3	2						
PANEL GLUE DESCRIPTION	A28088301	3	1	paper					
PANEL GLUING PROCESS	555124400	3	1	process??					
HOTMELT SALTABOND 9070	330619201	4	0.001						
SILIC DOW CORNING EA2626	397092201	4	0.095	Adhesive (silicon)	N-olefins {GLO} market for Cut-off, U - Waste type landfill				
SILIC DOW CORNING Q3-3636	358199056	4	0.0023	Adhesive (silicon)	N-olefins {GLO} market for Cut-off, U - Waste type landfill				
PANEL GLUING PROCESS	555123301	3	1	process??					
SILIC DOW CORNING Q3-3636	358199056	4	0.0023	Adhesive (silicon)	N-olefins {GLO} market for Cut-off, U - Waste type landfill				

Component	Code	Lvl	Qty	Material	Material (Ecoinvent)	Main process	Main process (Ecoinvent)	Mass (g)	Notes
SILIC DOW CORNING EA2626	397092201	4	0.095	Adhesive (silicon)	N-olefins {GLO} market for Cut-off, U - Waste type landfill				
LUX CPL. CL	A25128530	2	1						
CONF. DATA VAR SPEC	S00008817	3	1						
LUX OUI MERGED FILE	S00009570	3	1						
LUX HW	A15331520	3	1						
LABEL CD BARRE EAN13	356610817	3	1						
TASMA AXR7+84X600	561242400	3	0.15	thermal transfert ribbon (google)					
KNOB PIPO JL	A13909805	2	1	plastic					
KNOB PIPO JL	A13935905	2	1	plastic					
METRIC RECE.SCREW M4X8-4.	387856208	2	4	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		
RESISTANCE SEL 10+0 DX	357084003	2	1						
SWITCH BITENCODER	809062602	2	1						
KNOB HOLE COVER	A06150005	2	2						
MACS WIRING MEC1	561094514	2	1						
LUX ENCODER WIRING	561224213	2	1						

Packaging

Component	Code	Lvl	Qty	Material	Material (Ecoinvent)	Main process	Main process (Ecoinvent)	Mass (g)	Notes
PACKAGING GN	A21359801	1	1						
BOTTOM CUSHION	A21361001	2	1						

Component	Code	Lvl	Qty	Material	Material (Ecoinvent)	Main process	Main process (Ecoinvent)	Mass (g)	Notes
SIDE CUSHION LEFT	A25325101	2	1	EPS	Polystyrene, expandable {GLO} market for Cut-off, U				
SIDE CUSHION RIGHT	A25325102	2	1	EPS	Polystyrene, expandable {GLO} market for Cut-off, U				
TOP CARDBOARD	A03805201	2	1	CORRUGATED CARDBOARD					
CORRUGATED CARDBOARD	A25134901	2	1	CORRUGATED CARDBOARD					
THERMOSHRINKING SPOOL	356310811	2	0.24	PE- Low Density	Polyethylen,low density, granulate {GLO} market for Cut-off, U				
ADHESIVE TAPE	561218400	2	1.5	Adhesive (silicon)	N-olefins {GLO} market for Cut-off, U - Waste type landfill				
GLUING DESCRIPTION ARIANE	555032802	2	1						
	DC04ED								
	Z100								

Other top-level items (not part of the seven named modules)

Component	Code	Lvl	Qty	Material	Material (Ecoinvent)	Main process	Main process (Ecoinvent)	Mass (g)	Notes
ELC ML 00	400504600	1	1						
SCREW 4,2X9,5 DIN 7981	387854510	1	2	Carbon steel	Steel, low-alloyed {GLO} market for Cut-off, U	Turning	Steel removed by turning, primarily dressing, computer numerical controlled {GLO} market for Cut-off, U		

