

# Interface paper

Version 1.0.1

May 2<sup>nd</sup>, 2017

Andreas Ciroth, Sebastian Schulz

[ciroth@greendelta.com](mailto:ciroth@greendelta.com) & [sebastian.schulz@thinkstep.com](mailto:sebastian.schulz@thinkstep.com)

## Content

1.	Motivation and background of the paper .....	3
2.	Common Features of LCA Software Tools.....	4
2.1.	Structure.....	4
2.2.	Processes:.....	4
2.3.	Process data set documentation: .....	5
2.4.	Exchanges and flows .....	5
2.5.	Parameters: .....	5
2.6.	Product system:.....	6
2.7.	Documentation:.....	6
2.8.	Nomenclature.....	6
3.	Specific features and properties of LCA software tools.....	6
3.1.	GaBi.....	6
3.1.1.	Unique exchanges required .....	6
3.1.2.	LCA models, plans .....	6
3.2.	- openLCA .....	7
3.3.	RangeLCA .....	7
3.3.1.	Life cycle stages in product systems.....	7
3.3.2.	Statistical distribution .....	8
3.3.3.	Reference to other exchange .....	8
3.3.4.	Parameters, scenarios .....	8
3.3.5.	No allocation rule parametrization.....	8
3.3.6.	Links between processes .....	8
3.3.7.	Exchange with openLCA .....	8
3.3.8.	Exchange with GaBi .....	8
3.3.9.	Unique exchanges required .....	8
3.4.	SimaPro.....	8
3.4.1.	Product stages in product systems .....	8
3.4.2.	No UUID in export available, processes link to exchanges by name .....	9
3.4.3.	Links from external excel sheets into the models .....	9
3.4.4.	Allocation and system expansion.....	9
4.	Modelling guidelines to ease model transfer from one software to another .....	9
4.1.	Naming convention for processes and life cycle stages: .....	9
4.2.	Modelling Processes: .....	9
4.3.	Flows: .....	10
4.4.	Parameters: .....	10
4.5.	Life Cycle Stages .....	11
4.6.	System: .....	11
4.7.	Documentation:.....	11
5.	The background data sets .....	12
6.	To fulfil the mandatory requirement, it could make sense to separate the foreground from the background system. Interface workflow .....	12
6.1.	Model transfer into openLCA and GaBi .....	12
6.2.	Import test.....	13

## 1. Motivation and background of the paper

The EF pilots are modelled currently in different LCA software tools:

- GaBi ([www.gabi-software.com](http://www.gabi-software.com))
- SimaPro (<https://simapro.com/>)
- RangeLCA (<http://www.rdcenvironment.be/index.php?page=services>)

These tools are again used for the remodelling of the pilots. Further, two PEF pilots are to be modelled in openLCA (<http://www.openlca.org/>).

While all LCA software systems are developed for LCA modelling, calculation and analysis, they are also different:

- a) they contain features not present in other software
- b) they pose specific requirements and apply certain assumptions which have implications on the modelling or documentation, especially in the sense of limitations

The EF models created in the various software systems are foreseen to be imported into openLCA and GaBi, which in turn are used to export those datasets into an Excel template (openLCA) and an extended ILCD format with the current working name “eILCD” (GaBi), with the other software tool (GaBi and openLCA, respectively) as fall-back option. From GaBi and OpenLCA, the aggregated datasets and also some exemplary full models are to be transferred into the following LCA software tools:

- GaBi ([www.gabi-software.com](http://www.gabi-software.com))
- SimaPro (<https://simapro.com/>)
- RangeLCA (<http://www.rdcenvironment.be/index.php?page=services>)
- OpenLCA (<http://www.openlca.org/>)
- Umberto (<https://www.ifu.com/en/umberto/>)

The goal is to have all EF datasets as aggregated datasets in all software tools and as well some of them as full models. All project partners agreed to test existing data exchange formats for aggregated datasets before moving on with an exchange of full LCA models. To do that all agreed to test the exchange procedure with one process from GaBi, SimaPro and RangeLCA in ILCD format, and import into openLCA and GaBi to check feasibility of the data transfer and to compare the results afterwards.

The exported pilot models need to be compliant with PEFCR guidance 6.1 and the corresponding OEFSR guidance and should ideally also be recognised by the Technical Secretariat (TS) of each of the modelled pilots.

The remodelled EF models will use the official EF background data sets for energy and transport, chemicals, textiles, and many other sectors and products. The majority of these data sets is currently created; for energy and transport, the data sets are already available. These background data sets have, potentially, own implications, which need to be considered as well in this paper.

Aim of this interface paper is to compile and analyse the different requirements and unique features of the four different software systems listed above, to identify:

- unique features and parts thereof (e.g. specific parametrisation aspects) and conventions that are to be avoided in the remodelling and modelling of the pilots, since they cannot be well reflected in the other software tools.

EF remodelling project

- assumptions in the software systems that need to be recognised and that are posing a potential limitation on data exchange and / or on meeting PEFCR/OEFSR requirements.

More specifically, the aim is to

- identify common features which are supported by all four software systems mentioned above, and which are promising to fulfil the PEFCR requirements and the TS expectations, and
- to indicate unique features which are to be avoided, and
- to identify requirements and limitations which are to be recognised.

For both unique features and requirements, workarounds are provided as far as possible.

This text is written in a dynamic project setting where several aspects are not fully clear yet which are likely to have an influence on the final decisions taken in the text. These currently not fully specified aspects include the following:

- the final PEFCR/OEFSR requirements are not clear yet
- the flow list, which is to be used is not clear yet
- details of the model exchange are not clear yet (need to actually exchange models and at which level - full models, simplified models, aggregated LCI data sets? not clear)
- the final “eILCD” format and a possibly modified ILCD folder structure format are not clear; and finally
- the Excel-based format is not clear as well

## 2. Common Features of LCA Software Tools

Understanding common and individual unique features in the different software tools help to derive a modelling guideline which aims to ease the exchange of models from one LCA software tool to another. This chapter aims to specify a “common ground”, common features and properties of the exported models which are to be fulfilled, and can be fulfilled, by the different LCA software systems mentioned in the introduction. These are addressed in the following by “the LCA software tools” or “all LCA software tools”.

All the aspects listed below are the authors’ assumptions and should be double-checked by the software providers of the respective LCA software tools.

### 2.1. Structure

All LCA software tools can export process data sets with exchanges (i.e. flows included in processes), with parameters and a documentation.

### 2.2. Processes:

The LCA software tools have in common that it is possible to define processes

- with process documentation,
- with input and output flows,
- with flow amounts, possibly additionally defined by parameter values and formulas,
- with flow amounts specified with a flow property and a separate or combined unit (e.g. 1 MJ LCV of flow X or 1 kg mass of flow Y).
- with documentation (meta data), more or less detailed and complete

In processes the following information shall be included:

EF remodelling project

- Default Allocation Method
- Allocation Factors
- Exchanges
- Location
- Parameters
- Process Type

### 2.3. Process data set documentation:

Each process shall contain the documentation according to EF requirements and also be compatible to the ILCD format.

### 2.4. Exchanges and flows

The LCA software tools have in common that they use flows, and flows linked to processes, which are called exchanges in ICLD format

- with flow properties like mass, energy, LCIA characterization factors.

Flows shall contain the following information:

- Flow Type: The type of the flow. Note that this type is more a descriptor of how the flow is handled in calculations.
- CAS: A CAS number of the flow.
- Formula: A chemical formula of the flow.
- Flow Properties: The flow properties (quantities) in which amounts of the flow can be expressed together with conversion factors between these flow properties.
- Location: Normally the location of a flow is defined by the process location where the flow is an input or output. However, some data formats define a location as a property of a flow.

### 2.5. Parameters:

Probably all LCA software tools have in common, that they are able to use parameters

- which can be used to define flow amounts in processes:
- which can use formulas
- which can reference other parameters (in formulas)

More specific proposal:

- The following syntax of equations in the parameter section is permissible:
  - parameter names of up to 15 characters
  - parameter names without spacers or special characters or mathematical operators (like + - \* / %<sup>2</sup> ^& “ etc.)
  - formulas with standard mathematical operations like addition, subtraction, multiplication, division, exponential and root calculation, if-statements,
  - references (in formulas) to other parameters but without cross-references (one process to another process, e.g.)
    - without circle references or divisions by zero

## 2.6. Product system:

Probably all LCA software tools have in common, that they can define an overall system:

- containing process chains or networks interlinked with flows.

## 2.7. Documentation:

Probably all LCA software tools have in common, that it is possible to document the overall system, in addition to the documentation of the individual processes:

- with a description of the system including system boundaries, system expansion, goal and scope, assumptions, data gaps etc.

## 2.8. Nomenclature

- elementary flows shall be fully mappable to the ILCD elementary flows (potentially splitting flows, e.g. an ammonium nitrate emission flow into ammonium and nitrate emission flows, or mixed ores into the elementary flows of the contained elements or minerals)
- only flow properties of the ILCD reference data shall be used
- only units and unit groups of the ILCD reference data shall be used
- only locations of the ILCD reference data shall be used

## 3. Specific features and properties of LCA software tools

Probably all LCA software tools have their individual features, which are unique and not represented in other LCA software tools. Similar to the list of presumed commonalities in chapter 2, here the list of specific features and properties. The specific features are described in the following per LCA software and in each case, a recommendation on how to deal with the specific feature in the modelling is provided.

According to the “architecture” of the LCA model creation, implications in openLCA and GaBi have influence on the other two LCA software tools used for the modelling.

### 3.1. GaBi

#### 3.1.1. Unique exchanges required

For one process, an exchange must not appear more than once on input or output side.

- indication: If EF models contain the same flow several times on input or output side of the same process, they shall be changed and renamed
- responsibility: LCA modeller

#### 3.1.2. LCA models, plans

GaBi LCA models are called plans, which can be hierarchically organised. At present, they can be exchanged in a proprietary, non-text format only.

- indication: GaBi modellers will create one model (plan) per representative product, which contains one (sub)plan per life cycle stage. The name of the (sub)plan shall indicate the life cycle stage it represents (according to the guidance document 6.1). Furthermore only 4 hierarchical levels of subplans are permitted. Alternatively and/or additionally, modellers can use the grouping function to document the life cycle stage of the processes in the model. It shall be considered in the eILCD format, how to transport the information on life cycle stages or subsystems.

One special feature of the GaBi software are plan-parameters. Those hierarchical plan-parameters shall be forbidden to decrease the complexity necessary to exchange the models until an eILCD

format exists where plan-parameters might be addressed. Process parameters are allowed. Parameter values shall be documented in e.g. screenshots of the model settings as well to enable manual quality checks of the parameter values after the data transfer.

The delivery of multifunctional processes is forbidden, as for other tools. Allocation that was used must be documented, as well as substitution / system expansion. The process shall be saved and transferred with the parameter values valid after the allocation. Only the “already allocated process” is transferred, so that no adaption is necessary after the data transfer into another software.

### 3.2. openLCA

openLCA currently has three main limitations referred to the PEF pilots, for processes:

- quantitative references can only be on the output side of a process
- only product or waste flow types can be quantitative references, this is not possible for elementary flows
- waste flows cannot be modelled using the physical flow direction but the service direction.

While all these limitations can to some extent be worked around, they will be overcome in a new development by GreenDelta. Since openLCA is not one of the “original EF modelling tools” (which are GaBi, RangeLCA, SimaPro), this new version of openLCA can be used internally first at GreenDelta and does not need to be publicly available yet.

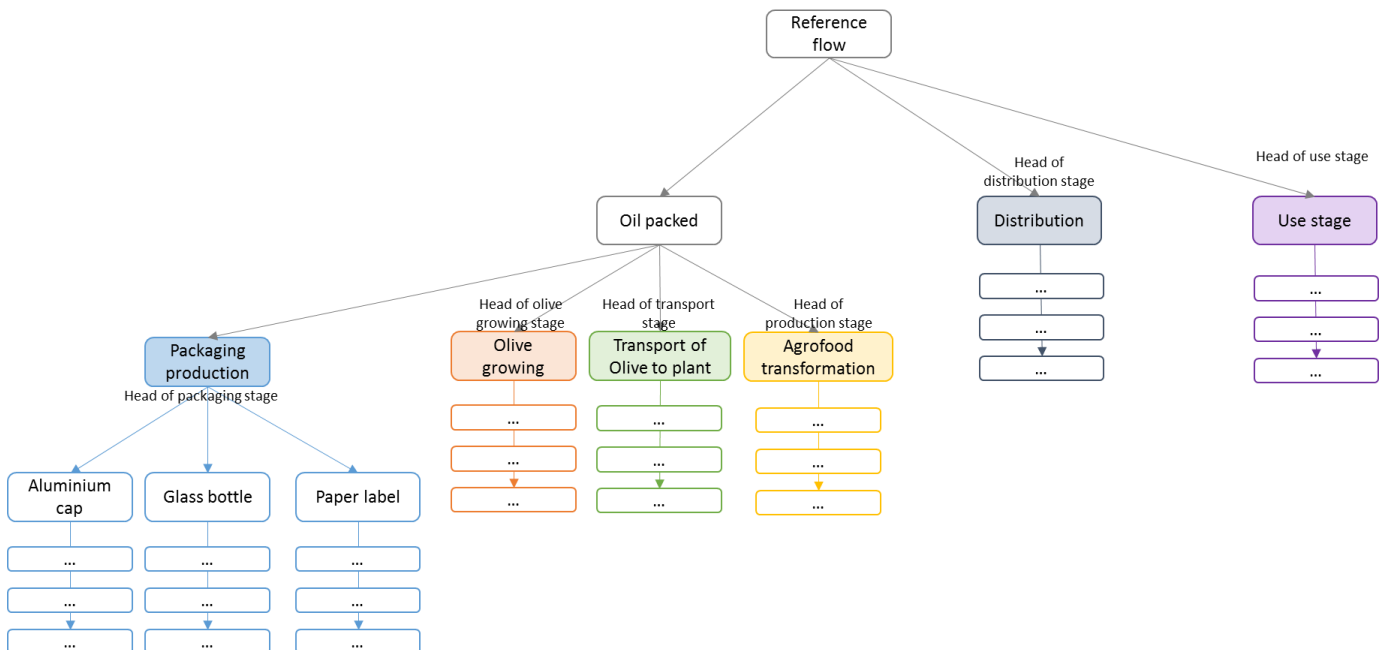
Further, openLCA does not contain product stages.

openLCA is used to export all models in Excel format. The workflow proposed is described in section 6.

### 3.3. RangeLCA

#### 3.3.1. Life cycle stages in product systems

RangeLCA models allow to create life cycle stages in product systems. A life cycle stage is defined by its “head” and all downstream processes linked directly or indirectly to this “head” process. (see picture below). All processes from a life cycle stage shall be linked by reference exchanges.



- indication: LC stages will be addressed by modification of the name of the process, the

name of the LC stage follows the PEFCR guidelines 6.1. The LC stage is added to the name of the process. The naming convention for processes and life cycle stages is further explained in chapter 4.1

### **3.3.2. Statistical distribution**

Possibility to enter statistical distribution as formula in exchange.

- Indication: avoid using statistical distribution for the remodelling of PEF models

### **3.3.3. Reference to other exchange**

Possibility to refer to a process exchange in another process exchange (exchange between process A and B is equal to exchange between process X and Y).

- Indication : avoid using reference to other exchange or convert to value when exporting in eILCD format

### **3.3.4. Parameters, scenarios**

Possibility to create different sets of parameters used in exchange formula, each of them being a “scenario”

- Indication: avoid using scenarios or ignore non-active scenarios when exporting in eILCD

### **3.3.5. No allocation rule parametrization**

Allocation rules are not parameters that can be changed but shall be selected when loading a specific database. The final modelling value will be stored and the allocation applied, if any, will be documented.

### **3.3.6. Links between processes**

Links between processes are stored in the process, and available for an export.

### **3.3.7. Exchange with openLCA**

For an exchange of models with openLCA, the existing ILCD extension format of openLCA will be investigated by RDC. The specification is shared by GreenDelta. The format already allows an exchange of models and is available in openLCA. Use of the format depends on timelines and effort, as fall back, the same approach as for the exchange with GaBi is used.

### **3.3.8. Exchange with GaBi**

The exchange with GaBi follows the “interface workflow” described in section 6.

### **3.3.9. Unique exchanges required**

For one process, an exchange must not appear more than once on input or output side.

- indication: Avoid

## **3.4. SimaPro**

### **3.4.1. Product stages in product systems**

- unique feature

- indication: product stages shall be avoided in product systems. Also waste scenarios (another unique feature of SimaPro) shall be avoided. Instead, process data sets shall be created.



### **3.4.2. No UUID in export available, processes link to exchanges by name**

- Pre indicated, that they are able to export flows with UUID's

### **3.4.3. Links from external excel sheets into the models**

Simapro can save parameters with links to excel spreadsheets, and update the parameters in the model once the parameter in the Excel sheet is updated

- unique feature

- indication: This feature is not supported by the other LCA software tools. It shall be avoided by the modeller to use those links to Excel spreadsheets.

### **3.4.4. Allocation and system expansion**

Allocation and system expansion are both supported and, if applied, documented in the processes. This is further explained in chapter 4.2

## **4. Modelling guidelines to ease model transfer from one software to another**

To be able to transfer LCA models (with all interlinked connections and values, formulas and parameter values, and others) from one LCA software tool to another, it is crucial to define commonalities of LCA models in different software systems (chapter 2). To avoid that model information gets lost during the data transfer, it is also important to define potentially unique model information and software properties, which cannot be represented in all LCA software tools (chapter 3).

In this chapter some modelling guidelines are defined to help modellers in various LCA software tools to focus on commonalities of all LCA software tools and to avoid using tool specific unique features.

### **4.1. Naming convention for processes and life cycle stages:**

To ease the data handling in the project, all consortium partners shall follow a naming convention for processes. The process names shall either directly carry the name of the life cycle stage (GaBi, RangeLCA), or it shall be possible to link the processes used in a product system model to a life cycle stage (Simapro, openLCA). The linkage is possible if for each stage, one process is identified as main process (or, the process is newly created) and the other processes in the respective life cycle stage are linked to this main life cycle stage process, directly or indirectly via a supply chain. The name of the life cycle stage follows the PEFCR guidelines 6.1, as follows: PEFCR life cycle stage 6.1 [alternative life cycle stage name used in the pilot]

To avoid technical complications when handling data in Excel or .csv formats, characters like comma, semicolon, slash, backslash shall be avoided.

### **4.2. Modelling Processes:**

When defining processes, each process shall:

- have a name limited to 300 characters (incl. spacers)
- contain flows only on input and output side
- contain a defined amount for each flow independent of accumulation/storage/time
- contain flow amounts defined with ILCD flow properties and units (as kg mass, MJ upper calorific value, pieces, m<sup>2</sup> area etc.)
- not contain information on flow amounts, which are only defined outside of the model (e.g. having a reference to an Excel sheet instead of a number defined in the process. Of course

interfaces to Excel or other software systems can be used to define the flow amount, but then the flow amount shall be saved in the process as a number as well.)

- contain the same flow only once per input and output side (e.g. NOT having the same electricity flow or resource flow multiple times on the input side)
- allocation should be avoided, if possible. Else it shall be documented in the foreground unit processes. The documentation must explain which by-products have been neglected and what is the considered main product. The allocation method and the allocation factor must be documented.
- multifunctional processes (a process with multiple product flows/functional units) is forbidden in all software tools in the scope of this projects. Allocation may be used to allocate the process inputs/outputs to one main product and to get rid of the by-product-flows. The process shall be saved in its state after the allocation, with only one product flow/functional unit and with already applied allocation factors to the other flows and without by-product-flows. The goal of this restriction is not to forbid allocation (it can be done before the data transfer), but to allow an easy transfer of such a process into another software. It shall be possible to transfer all processes “as is”, without the necessity to apply changes to it afterwards.

#### 4.3. Flows:

- Used elementary flows shall be fully mappable and mapped to the ILCD elementary flows (potentially splitting flows, e.g. an ammonium nitrate emission flow into ammonium and nitrate emission flows, or mixed ores into the elementary flows of the contained elements or minerals). The pilot remodelers are responsible for using the ILCD flows or for mapping their flows to ILCD flows to provide exported datasets with ILCD flows. In case of an automatic flow mapping the software vendor is responsible for the mapping.

#### 4.4. Parameters:

If parameters are used in the model:

- Avoid parameters which reference to other parameters outside of the model (e.g. global parameters, project parameters, parameters in other software or external databases etc.).
- Parameter relations from parameters used in a process to parameters outside of the model will get lost, when transferring the whole model from one LCA software tool to another. It is in the responsibility of the modeller to avoid this.
- If values of process parameters are dependent on other parameters outside of the process, the value of the parameter shall be documented or fixed, so that the value of the process parameters do not get lost or can be redefined in case the parameter relations are not transmitted during a data transfer from one LCA software tool to another.
- The following syntax of equations in the parameter section is permissible:
  - parameter names of up to 15 characters
  - parameter names without spacers or special characters or mathematical operators (like + - \* / %<sup>2</sup> ^& “ etc.)
  - parameter names that do not begin with a digit (e.g. names like “2parameters”)
  - formulas with standard mathematical operations like addition, subtraction, multiplication, division,
  - references (in formulas) to other parameters

- without circle references or division by zero

#### 4.5. Life Cycle Stages

The life cycle stages in an LCA model are an essential information, which needs to be transferred. The information on life cycle stages is implemented differently in all software tools. For GaBi, in the documentation of each unit process the life cycle stage shall be documented in the documentation field “model description”, following the LC stages name used in the PEFCR guidelines 6.1. For SimaPro, openLCA and RangeLCA, the life cycle stages following the PEFCR guidelines 6.1 shall be created as processes, and the processes in each respective life cycle stage shall be linked to this life cycle stage process. A screenshot of the structure shall be provided to enable manual quality checks of the model after the transfer into another software. It must be possible to connect the life cycle stages with process name, as described in chapter 4.1

#### 4.6. System:

Transferring very complex LCA models from one LCA software tool to another might be challenging. Because of this it might make sense to consider modelling only as complex as necessary.

- If in the scope of the remodelling tender models are to be transferred, the models shall be modelled with in a simplified way using only the described software features described in the interface paper.
- Scenario modelling (i.e. having one LCA model with several different scenario parameter constellations, where it is not clear, which parameter settings are the relevant ones) is to be avoided, since the scenario information might get lost during the data transfer or because not all LCA software tools might support scenario modelling.
- One model shall represent one product.
- If one model represents in different (scenario) variations several datasets, one copy of the model with unique settings shall be saved for each dataset.
- All numbers in a model shall be saved inside the model to be transferrable with the model. There shall be no references to external data sources (e.g. global parameters in other databases or references to external files etc.)
- In every system it shall be clear which process delivers the functional unit.
- The system shall be scaled to the functional unit.
- It shall be avoided to use grouping of processes or graphical elements as colors or pictures, if this is result relevant and results are changed when this information gets lost during data transfer.

#### 4.7. Documentation:

The whole system shall be documented not only according to EF requirements, but also in a way that it enables others to rebuild the model in case part of the model information got lost during transfer. This could be:

- goal and scope
- the functional unit and which process delivers the functional unit
- allocation and which processes use allocation and how the flow amounts inside those processes are affected by allocation
- information on system boundaries and system expansion

- fixed parameter values throughout the system for all processes, independent of their references to formulas and other parameters.

The most practical approach to test the data transfer in the scope of this project is to deliver screenshots of the models, which allow to reconnect the process chains by hand, if necessary.

## 5. The background data sets

As mentioned in the introduction, most of the background data sets are currently created, therefore it is not possible to have a full overview of their properties.

In principle, the data sets are to be compliant with the EF rules, but they may deviate for two reasons:

- the data sets are referring to a different version of “EF compliance”, as specified in the respective data tenders and thus are not fully consistent, while it is still to be evaluated in how far this would affect the data set exchange/use
- the data sets differ for other reasons, particularly due to some possible lack of fixing conventions that can mean different data set providers use their own, distinct conventions.

For the remodelled pilots, it is mandatory that the foreground system is compliant with the PEFCR version 6.1 and the corresponding OEFSR guidance and where possible in agreement with the respective TS; it is not, strictly speaking, required that the entire model including the background data sets is compliant. Background data sets will be used by the project “as is”; any non-compliances are outside the project’s responsibility. If however, the data set causes issues in model transfer, this is to be solved.

## 6. To fulfil the mandatory requirement, it could make sense to separate the foreground from the background system. Interface workflow

After the modellers have remodelled their models with EF compliant datasets, the models need to be transferred into the openLCA and GaBi software, as prerequisite to achieve deliverables 1a and 1c. From openLCA the models are exported in an Excel format (task 1a), while from GaBi the models are exported in eILCD format (task 1c), to be imported again, at least for test purposes, into all other software tools in eILCD format (task 3).

For task 1a and 1c it is required, that the RP-models are provided by the modellers with a correct ILCD flow list, since only then they can be imported correctly, which is a prerequisite to export them again properly.

### 6.1. Model transfer into openLCA and GaBi

There are 2 options to achieve the data transfer from the origin software, where the first versions of the models were modelled, into openLCA and GaBi:

1. eILCD as “common interface format”

The eILCD format needs to be defined in the scope of the project. It would be beneficial for the data transfer if this format is available in all software tools already at the beginning of the project, since it allows the transfer of whole models. A data exchange with the eILCD format would prevent manual efforts for the data exchange or decrease it drastically. However, the eILCD format can only be used for the import into openLCA and GaBi, if all software vendors implement an eILCD export and if an eILCD import is available in openLCA and GaBi.

2. If no eILCD format is available for the data exchange, the models have to be brought into openLCA and GaBi with existing data exchange formats (“best of each software”).

Here, from each software, the format is used which exists already and which best meets the requirements specified in this text. According to the discussions so far, this would be:

To \ from	GaBi	RangeLCA	SimaPro
openLCA	ILCD*	ILCD**	CSV
GaBi	-	ILCD*	ILCD***

\* with additional information for process connection

\*\* probably an extended format that stores connections; otherwise original ILCD

\*\*\* according to PRé, export is only available for system processes; according to own tests, the export seems to work also for unit processes; if the public version of SimaPro allows only system process export, then a specific exporter will be provided by PRé, or the ILCD datasets will be provided by PRé, created by this exporter

Transferring a whole model in the original ILCD format is not possible. If the ILCD format is used, all processes of a model are transferred, but the connection between the processes gets lost. In this case, all process chains/the process network needs to be reconnected in openLCA and GaBi. In this case the remodellers have to be responsible for the quality and consistency of their models in openLCA and GaBi, also they are responsible to do the transfer of their models into the 2 software tools. GreenDelta and thinkstep will provide a free version of openLCA and GaBi to all modellers for the duration of this project to do that, if necessary. Additionally, thinkstep and GreenDelta will provide an EF compliant database with the background datasets to facilitate the modellers to import their models. The free license of GaBi will be available once the EF- database is available and the import tests have been concluded successfully. Then it is only necessary to transfer the foreground datasets into the GaBi and openLCA database and to reconnect it with the EF-datasets. GreenDelta and thinkstep will deliver technical support to the modellers to make the best use of the software and the existing import interfaces.

After the modellers have imported their models into openLCA and GaBi they hand over the openLCA database to GreenDelta and the GaBi database to thinkstep. Both GreenDelta and thinkstep can support the modellers with a quality check of the models during a handover meeting.

If it seems easier for the remodellers to do the remodelling in openLCA or GaBi from the beginning, they can also import their models into openLCA and GaBi before they start the remodelling. Then they exchange their background datasets with the EF compliant dataset in GaBi or openLCA after the transfer of the model.

For the remodelling the background datasets have to be substituted by EF compliant datasets. To do this the EF compliant background datasets need to be imported into each software tool first to enable the modellers to bring them into their models. To do that the following procedure is agreed upon:

## 6.2. Import test

1. Import one EF compliant dataset in ILCD format into each software to test the ILCD import. If no ILCD import interface is available, this needs to be done with a workaround, e.g. using a format converter or using openLCA.
2. Every software vendor calculates LCIA results according to the latest PEF recommendation of LCIA methods and exchanges the result in an Excel file with the other software vendors.
3. The LCIA results for the various dataset are compared. Differences need to be discussed.
4. If the import of the exemplary EF compliant dataset was successful (same LCIA results in all software tools), all other EF compliant datasets of the EC shall be imported via the ILCD

format into each software tool.

5. After the import of all EF datasets another quality gate is needed to ensure, the background EF datasets in every software tool produce the same results. If this fails, it does not make sense to go on with the remodelling or to test the LCIA results after an exchange of whole models.
6. After it is accomplished to have the same background EF database in every software tool the modellers can start remodelling.

The import test should ideally be performed for several data sets, in different levels of aggregation.