Supply Chain Sustainability Footprint

Fundamental Concepts v1.0, Lorand Kedves, 2025-01-31

Abstract

The goal is to create an information system collecting relevant data, executing custom calculation of various sustainability-related indicators, and publishing them attached to products sold to external customers.

The pilot version focuses on nitrogen footprint management at a farming community, but the solution should be open for extensions. The university will work out measurement methods and calculation formulas, the farmers will provide the data to the system running at the community center that collects, stores and sells the goods.

This document is a short summary of fundamental concepts, all terms and statements are open for discussion.

External Context

Reporting entities must disclose various sustainability indicators related to their activity, both for their organization as a whole and by their products. The values must come from verifiable input data and transparent calculations, open for any later audit.

The entities are responsible for collecting all relevant data and manage their formulas describing their operations. However, there are intermediate products that they receive from external entities. For such items, their producers must provide these indicators in a similar manner. We will call this Product Sustainability Footprint Data (PSFD).

A farming community produces agricultural goods like grains by using electricity, fuel, fertilizers, pesticides, etc. The buyers of the grains need these indicators, either by “units” like an identified bag of grains, or by measurement unit (a ton of grains).

Abstract Entity and Activity Model

A farmer community is a cooperation of farmers. A farmer executes various activities on one or more fields. The system must record all activities that affect sustainability indicators by field. In simple case, one field grows one product at a time. If there are more products on the field like companion planting, optionally with different harvesting time, the calculations are more complicated but can be managed, either by registering activities by product or distribute the values on harvest.

After harvesting, the product may be stored as a whole or put into individual identifiable bags. The PSFD is calculated from all recorded activities on that field and assigned to the product, distributed by the mass / volume, by unit or by the capacity of the bags individually. At this point, the farmer and the field is identifiable.

Products can be mixed even by opening and adding the content of the bags. Mixing results averaging the PSFD and stops unambiguous source identification. Still, it can contain the ratio of the source field and farmer (may be interesting in regulatory investigations).

Fields may have special flags like “permaculture” that are inherited to the products and limit the activities allowed there. These flags are cancelled when mixed with non-flagged products.

Products can be transported to the community site. In this sense, the site is another entity, managed by the people assigned to it (not the farmers). The transported goods have their current PSFD, can be mixed or stored separately. Any activities at the site (like, drying, heating, relocation) are similarly recorded in the system and attached to the PSFD. When selling to customers, the final PSFD is calculated and attached to the goods the same way as explained before.

Technical Operations

We assume that the community must publish a sustainability report each year, while outgoing PSFDs are created and sent to the customers continuously. The system should attach digital signatures to all outgoing data and keep them for later audits.

The regulators expect sustainability reports in XBRL format, using the ESRS taxonomy in the EU. The calculations must include the PSFDs of the incoming products. The easiest approach is to store PSFD also as an XBRL instance and ESRS taxonomy, therefore no custom conversion is required anywhere in the supply chain.

The core problem is that the PSFD input data (like the official footprint of energy or a specific fertilizer) and calculations will change over time, while reports and PSFDs once sent are final and should be auditable.

To deal with this issue, the information system does not represent a “current state” of the system but collects events (activities on the fields) with the granularity (activity type, used material type and amount, used fuel, etc.) required at the current moment. The calculation package allows using the data from a selected set of events (all activities from preparation to harvest) to produce the current amount of goods and generates the PSFD. This package is also final and will be accessible with its validity period from activation to deprecation. To change it, the administrator clones, updates and activates the new version (that automatically deprecates the current one).

The main activities are the following.

Organization Management

The community administrator registers the farmers, fields, products, etc. This is an ongoing maintenance to keep the system in sync with the organization. The system operates following this configuration, like farmers can log in, access their own fields, report their activities.

The system may contain a limited warehouse management or connect to an external system if present at the community. This lists the storage facilities (silos, cold storages, treatment sites, etc.) both the community site and at the farmers. This is where mixing can occur, or PSFDs are updated with the footprint of a treatment. Also, this system provides unique identifiers to “bags” if that service is needed.

The organization setup data is considered "mutable"; if anything should be audited later, it is copied into the Reporting Period.

Reporting Period

All data is collected by reporting period, assuming one year, by the current date. The collection is closed and finalized when the period ends, a new period is opened by copying the configuration (but not the data) of the closed one.

Operational Configuration

The period configuration is considered as "operational data" that allows registering relevant events. It contains items like:

- All values that should appear in the PSFD – mostly a selected from the ESRS taxonomy, but may contain items from the local extension taxonomy, to publish specific values to the customers that are not (yet) supported by the ESRS. A custom taxonomy is necessary, containing the fields and optional extra flags.
- All chemicals that are available at the community and can be used on the field. They must be registered before first using it as an administrative task, in cooperation between the farmer (who just bought it) and the community administrator, "admin" (to ensure that the type is registered only once).
- All attributes like fuel consumption, applied chemical type, applied chemical amount, etc. that the admin wants to collect for the activities.
- All activity types that can be executed on the fields, managed by the admin exclusively. The type also contains mandatory and optional attributes, even limitations (like, a certain activity can only use a given set of chemicals, or even "realistic amounts") to minimize input errors.
- Optionally: equipment types that can be used for certain activity types, or even instances. This may help finding outliers like a defect harvester by the strange fuel consumption – but means additional administrative task for the farmer. Thus, this is just an illustration for the pros and cons of increased granularity.

Calculation Package

The calculation package contains

- Optional temporal attributes that are calculated from the input data, to be reused by other calculations. Typical example is a "homogenization layer" from the direct, local input attributes that may be custom for products, to the same generic one that are used to calculate the target ESRS values.
- Expressions for direct calculation of attributes (like Excel cell formulas).
- Scripts when different expressions must be applied to calculate an attribute group depending on some complex conditions (like Excel macros).

Events

The main task of the system is to capture all events that happen in the community and has any kind of sustainability effect, in the required granularity and in its pure form. At this moment, there is not plan for inventory management: 50kg Fertilizer A in a bag in a warehouse does not change the footprint of the grains. However, the footprint of 10kg Fertilizer A applied on Field X must be distributed over the grains harvested from Field X.

The key here is that this information is locked to the field and not the product. Later analysis can investigate how much of its content went into the grains; more elaborate calculations may consider all events from previous years on Field X with decreasing percentage. The same event source can be evaluated through different calculation packages either to improve them according to the experiences, or to correct previously published reports or PSFD values in critical situations. For example, examining a sample shows too much of a chemical due to treatments from the previous years; with the corrected calculations you can look for other fields where the same situation may exist to test, call back or change classification of the products.

Field Treatments

Farmers report this event: they select one of their fields and the treatment type. The field may have special flags that either filter the selectable type or gives a warning if that type is not recommended / allowed. After selecting the type, the list of applicable equipment, chemicals, etc. appears, select the actual ones. Probably it is better to do this when starting the activity, as it is possible that multiple activities start at the same time. The activity is closed by providing the used amounts (fuel, chemicals, etc.) and submitting the event.

Technically, the activity log remains on the farmers' device or client application as well, they can review it anytime. Also, they may have no connection to the server from the place they work, thus the client application must be able to temporary cache the new data.

Sowing

This is a special treatment when a product is assigned to a field. The grains may have come from an external supplier, in this case the incoming PSFD is assigned to the field. Again, question if multiple products can be assigned to the same field? For the long run, the system should rather prepare for that.

Harvest

With this event, a measured amount or number of bags of the harvested product are created in the system. At this point, the calculation package is activated and the PSFD values are calculated, aggregating all events from the previous harvest (including any field treatment before sowing).

If there are multiple products assigned to the same field, the calculation package either handle this situation, or a specified proportion of the PSFD values are applied. The distribution is logged in the activity event and applied when the remaining products are harvested from the same field.

When harvested "en masse", it must be placed to a storage.

Store

We assume that any store can contain only one product. Adding to an empty store is simple, but if that already contained some amount of the same product, they are now mixed: the current and the added PSFD values are averaged by the amount. The PSFD of the incoming product is deleted.

Transfer

The store amounts are updated accordingly. The source store PSFD remains the same, the target PSFD is updated according to the Store operation.

Sell

The outgoing shipment is created from one or more store, their amounts are updated, PSFDs remain unchanged. A new public PSFD is created with a list of Store operations from each source by the received amounts. This PSFD is digitally signed, uniquely identified and archived together with all shipping information.

Transaction

One transaction may contain multiple shipments (different products in multiple trucks, containers, etc. together with bags of products as well). All PSFDs should be sent to the customer and/or become available for them on the community portal. It is also possible that they should digitally sign and send back the PSFDs to the community to close the transaction.

Implementation Considerations

This section assumes that the system is implemented based on the XBRLDock platform, created by the research network.

Deep but Partial XBRL Integration

The fundamental problem of this system, its continuously changing nature. We can be sure that the PSFDs should contain different fields, therefore, we should collect different data from the incoming products like chemicals. There will be new activity types, new calculations, etc.

The good news is, XBRL addressed exactly these issues, 20+ years ago and came up with a solid answer (on the abstract level), this is why the standard is the same since 2003, used for the most important financial reports in the US since 2008 and in the EU since 2023 because the reported content follows all changes every year. However, the standard itself, with its complicated XSD/XML configurations and report formats, is complicated, not really integration friendly.

The platform uses the relevant improvement in XBRL, like configurable and flexible attributes and types, configurable expressions and scripts. The primary storage format is simple JSON (hierarchical text). The necessary import from the official ESRS taxonomies is provided by the platform, the administrator only deals with the simpler formats. The platform also generates the standard XBRL taxonomy and instances (outgoing reports and PSFDs) from the configurations and data collected for the reporting period or aggregated events through the calculation package.

This approach ensures a fully XBRL compliant interaction with regulators and customers, while the internal data management and interaction does not require the complex technologies of XBRL (better performance and transparency) yet inherits its flexibility.

Simple Data Storage

The XBRLDock platform was implemented with academic research focus, it had to handle huge amounts of data stored in XBRL format (e.g. 260GB of 9260 real life reports from the EU ESEF program, or over 100 million data points collected by the SEC). For on-line analysis, the platform pre-processes the reports and store their content in simple text files (JSON and CSV).

The system will reuse this storage. The activity information is stored and can be handled just like a valid XBRL instance, but its total data content is accessible through a CSV file. All the rest is in JSON files in a simple directory structure. This allows very simple and transparent data management, backup and archival, creates zero dependency on any technology or tools.

Digital Signatures, Zero Trust

I any digital environment, it is very important to verify the sender and content, in this case, reports and PSFDs. The globally accepted standard solution is digital signature using the private/public keys of the entities. In short, a cryptographically strong algorithm creates a signature to a file using its content, the actual time and the sender's private key. The signature can only be opened by the sender's public key (ensuring that it really comes from the sender), contains the timestamp, and the same algorithm only gives the same signature to the file if the content is the same.

When publishing a report or a PSFD, this digital signature is added to it (the XBRL standard contains this service). The system may also support mutual signature as described in the Transaction section – this would ensure that the customer saw the same content as it was published.

It is possible to create public, independent, mutually trusted, read-only "vaults" where any party can upload these signed packages to resolve misunderstandings.

Improvement Opportunities

The above detailed, very flexible architecture allows significant extensions without compromising the data management layers, here are some of these opportunities.

Deeper Regulator and Customer Interactions

The core data model is very simple: activity events on the fields and PSFDs for products; both with configurable attributes. They are stored in a simple format but can be seen as XBRL instances using the automatically generated taxonomies. Some examples of the benefits:

- A simple, standard XBRL interface is enough to trace back any product or statement to its origins, to the first preparation activity on the field that later produced the content of the bag that I see. Similarly, most of the audit can use the same interface to collect and verify reported data, look for irregularities (like, strange amount of yield, fuel consumption, etc.)
- A customer may ask for a product with a special treatment, either by existing protocol at the farmer, or a new, custom method known to the customer (in

which case, the admin creates this new, custom flag). The farmer assigns this flag to the field, and on harvest, creates the requested number of bags that holds this flag, while mixes the remainder. The customer can experiment with the result, can ask for more or switches to another method.

This requires zero custom development, neither to create nor to keep it in sync with the changes, only administering the configurations.

Geoinformation

For farming, geoinformation can be important, to be assigned to the fields. This can either be managed in the platform or regularly updated from an existing system. Due to the configurative nature of the platform, this extension is relatively simple. The new information can be opened to the calculation package to verify mandatory distances, similar local climate, etc.

The recommended format is the existing standard GeoJSON that allows using existing services (like, display on a map) or publishing to external parties.

“Sensor Fusion”

It is very likely that over time other events should be added to the system, like temperature, rain, sunlight – or complex ones like a soil sample analysis. The results can be stored as a “measurement event”, with or without sustainability footprint to the field.

The collected data can lead to much deeper understanding and planning of any activity – again, without any change in the underlying storage and communication services, this is just another configuration change.

The geoinformation package allows the measurements to be assigned to a precise location within a field – or even, controlling active measurement devices like wheeled or flying drones equipped with sensors. Of course, in this case the map should be extended with tracks to follow or zones to avoid, and dynamically used by the drones to ensure safe operation, avoid collision, etc. This use case also enforces using the GeoJSON standard as any device is very likely to support it.

Next Steps

1. Discuss any question regarding to the plan, especially the core data model.
   Result: an agreed architecture.
2. Collect information for a pilot: attributes, products, activity types, PSFD content with their descriptions.
   Result: core services (data storage, generated user interfaces, calculation package, taxonomy and report generation) configured and verified.
3. Collect some numbers: farmers, fields, materials, products, etc., including assumed activity counts.
   Result: a virtual replica with realistic content or even an automated demo of activities.