MarketCommons functional specification for a sustainable economy database – Part 1

– A steadily growing document of basic concepts, requirements for the database, and –in the future– a functional specification of the Connected Web Applications

Version: 0.037
Date: Apr 17, 2017
Author, copyright: Oliver Sendelbach (Email: socialmedia (@t) ourconomy.com)

Table of Contents

1 Project description

This is the functional specification for a system that supports users in managing data about available economic resources. The aim of the combined system is to create the market information commons for sustainable supplies and providers of these supplies. It also wants to connect the actors of a sustainable economy. To this end, the design of a combined system of a central database and connected web applications is proposed in this document. The combined system is intended to serve as a technical backbone for a number of on-line communities working on sustainable business practices. Their supporters can feed static data into the shared central database and therefore can avoid duplicating their efforts.

'MarketCommons' has been chosen as a working title as a play with the two concepts. The data treasure created is the commons of market information.

1.1 Preliminary remarks

This document will evolve according to the idea of the 'minimum viable product'. Therefore, in the beginning it will describe only the most important and most urgently needed functionalities. They constitute already a working system. Along the path outlined below, it will develop into a more powerful tool with more convenient features.

In this document
Important items that have special meaning in this document will be written in capital letters, from this point on. Because this helps readability without being very beautiful, it is not used for user communication, except in variable names.

Building on existing work
The basic steps for the development of the Combined System consisting of the Central Database and the Connected Web Applications start, of course, with the Central Database. A basic Central Database already exists with Markus Kohlhase's OpenFairDB (http://www.openfairdb.org**).** Unfortunately for the author, he is a quick thicker and programmer so the document will run after him a little instead of giving him guidance for his development. It will nevertheless try to get some features into his progressive work.

1.2 The steps towards a full-blown system

Step 1: Data structure definition
First, the initial milestone of this functional specification will be the descriptions of the businesses JSON file and the products JSON file, along with explanations of the rationale behind them. In the current version of OpenFairDB a JSON description of a combined Event/Business/Initiave item exists already, but the products data structure will be really something new.

Step 2: Product and Service functionality
Second, there will be a specification for products management in a Connected Web Application. Data entry is done on web sites linked to the Central Database. For OpenFairDB, the first Connected Web Applications already exists: https://www.kartevonmorgen.org/ written by Markus and run by the initiative 'von morgen'. The specification will contain the product data entry page, the product search

page and the product information page.

Step 3: Valuations
The next, the third, step will be about the valuations. The document will contain two additional tables about valuation categories and valuation that can be given to products and businesses in these categories. The challenge will be to allow every Connected Web Application to offer their own valuation categories, but also to keep the system open to allow for usage of valuations of other Connected Web Applications that are considered useful.

Step 4: User registration
Fourth, the next step of this document will make suggestions about the treatment of users, registered users, and people responsible for businesses.

Step 5: Cumulated effects
Fifth, the next version of the specification will make suggestions about the aggregation of certain items in the supply chain of a given product, i.e. how much CO_2 is contained this product? How much virtual water has been used to make it? Regarding valuations, first thoughts and ideas about inheritance of valuations will be made. This will be a starting point for further research by other people rather than a finished functionality. Also, a JSON description for the data structure for managing hyper-links will be provided.

1.3 Important setup in analogue world

Apart from the code, the administration of the actual system will be organized to be self-managed by registered users and passers-by. A suitable consensus-democratic management system to achieve that needs to be implemented, but is out of the scope of this document.

2 The main elements: Places and effects

This is the basic foo of the document:

The subject of the market information commons can be described entirely by places and effects:

2.1.1 Place

Every place where something happens is a place, ie every shop, facility, event hall, venue, point-of-sale is a place.

2.1.2 Effect

Everything that happens or is produced in a place is an effect. All good things – products, services, natural resources – and all bad things – pollutions, scarcities, whatsoever – are effects.

2.1.3 Relations between them

Places can be nested in other places. This helps to locate Places.

Effects are usually composed of other effects. This very simple principle will hopefully help the users to see through the complicated supply chains in today's world economy.

The effects that are available at a place can be part of the dataset of the place. Effects contain the ID of their producers.

Only places have a (geo-)location. Effects currently do not have this as they are normally linked to Places. The project is open for user input in this matter.

2.2 Use of words

This document uses the terms 'business' and 'economic' for simplicity. Because some people are allergic to these and related expressions, it is important to stress that when 'business' is used it also means all sorts of initiatives, community centers, for-profit-businesses, free shops – all that you can imagine where effects are exchanged. 'Economic' is used much in the original sense of the word which comes from the ancient Greek word for 'household'. So all the places and effects that affect consumers' well-being are part of the economy whether they are connected to financial flows or not.

2.3 Tables

The main contribution of this first part of the functional specification are the tables for Places and Effects. The next sections will describe the main ideas behind both tables and, where needed, the meaning of theirs rows as well as their columns. The format of tables has been chosen to present the data structure in this text. In the code they will be represented by JSON files.

In order to express the existence of embedded tables in tables, the connotation “1...X” is used in the table templates. Where applicable in the respective rows of the table templates, references to master fields in other tables are given to avoid duplication of information and to make relations. This is explained in a section below.

The table templates can be found in the file MarketCommons_Database_vX.XXX.ods

3 Features of both the 'Places' and the 'Effects' tables

The tables in the functional specification share common elements. Especially, the Places and the Effects tables share the full set of common features.

3.1 Database description table rows

All the table templates share the same structure of rows. It is explained below.

3.1.1 Field category

The first row 'Field category' shows to which category the database fields belong to. The database fields are shown as columns of the tables.

technical: These data fields are technical fields that every entry in every table needs or could need.

metadata: Interesting data about the record entry but not the actual content of the record.

common: These fields are used in all tables if applicable. They contain information that is interesting in the records of all tables.

special: These fields apply only to the content data stored in one of the tables. They are different in each table.

3.1.2 Field implementation

This row contains information about the actual time when this field will be implemented for use by the Connected Web Applications.

3.1.3 Central Database requirement

This line contains things the Central Database needs to be able to do to work well with the Connected Web Applications.

3.1.4 Connect Web Application requirement

This line contains hints towards the minimal set of what Connected Web Application need to be able to do for compliance with the Central Database.

3.1.5 Explanation

This information is for the developer about meaning of the fields. It can also be used in the web application as “tool tips” to tell users how to fill the fields. Tool tips in other languages are needed at a later stage in the development and should be linked to the language management of the Connected Web Applications.

3.1.6 Field name(s)

These are the field names of the database. They are unique throughout the database.

3.1.7 Field type

'Field type' can be either “single value” or “table”. This is why the 'Field name' rows in some columns contain more than one field name separated by “|”.

This documents assumes a NoSQL database as the backbone of the Central Database. Field type 'table' means that a small nested table is needed, but not a fully featured main table as the Effects table, for instance. The structure should be usable without any problem for the Neo4J graph database that

OpenFairDB employs.

3.1.8 Reference to master field in other table

In order to avoid duplication of information, some fields can or have to contain references to fields in other tables. Wherever a field name is given in this row, a value from another table needs to be provided. The Connected Web Application needs some functionality to allow the user to search and choose these entries and establish proper relations. Sometimes, as for example in the component list, references are made to other fields in the same table.

3.1.9 Variable Type (including possible values)

This table row suggests a variable type for the variables named in the 'Field names' row: string,double float point, datetime, long integer, boolean, memo field.

They are subject to change in case of better suggestions by the database developers.

In case a field can only be filled with a pre-agreed set of values this is stated in the 'Variable Type' line in the field description tables.

3.1.10 Field optionality

This row says whether a field needs to be filled or not during data entry by a user. Connected Web Application need to mark these fields in their forms.

The next sections contain explanations regarding the data fields of the tables.

3.2 Explanation of database table columns: Data fields of the Places table

The data fields of the Places records are represented by the columns of the table template.

Please read the following sections in tandem with the table template file. This document only contains information where the Explanation row does not provide self-explanatory information.

3.2.1 Field category 'technical'

These fields provide some technical functionality for use where the database does not already have these features aboard. For more information on each column, see the 'explanations' in the equally-named row.

PlaceSubSysCustom fields
The fields of this embedded table are very versatile and can be freely used by the Connected Web Applications for uses that have not yet been implemented in the Central Database.

3.2.2 Field category 'metadata'

These columns hold useful data regarding the record, but not the actual informational content of the record.

PlaceModifierUserName
The database should only start to collect and display back these once users can use pseudo-anonymous user names of their choosing.

PlaceRecordStatus
Place records entered by users have the status 'contributed'. The community self-managed administration of a Connected Web Application can set a status to 'supported' as a kind of saved revision. In case a record is often vandalized by competing interest groups, these supported entries can serve as a source of trust.

3.2.3 Field category 'common'

Fields in this category contain the content of the record that all or most of the tables have in common. There is one important thing to note about the information given in the fields in category 'common' and 'special': Data entered in the categories 'technical' and 'metadata' are used in the background as a log on how the data was gathered. In contrast, newer entries to 'common' and 'special' fields enhance missing field values or supersede older field entries and constitute the combined record that is displayed to the user.

In order to display the current set of data correctly the Central Database needs to be able to distinguish between whether the value in empty fields was 'deleted' or 'never filled'. When a user does not enter a new value in a field a former value is taken. When a user deletes the former value in that field it must be considered as blank.

PlaceIsDoubleOfID
Users can use this field to indicate that there is a better entry for the same item with another ID. Combined Web Application can use this information to guide users to make their entries to the main dataset.

PlaceType
This functional specification suggests the following set as possible values for PlaceType:

Shop: Place where consumers can get supplies. The word “shop” is only a simplification and does not mean that one has to pay for the supplies. A freeshop can be a “shop”.

Commons: Natural or social resource that one can use in coordination with other members of the society. Example: Public park, where one can picnic and have a barbecue, or: the website Wikipedia.

Regulatory Body: All initiatives, NGOs, governmental agencies that help to guide economic activity in a sensible way. It is important to stress the role of civil society organizations in this area.

Producer: Places that are not “shops” in the sense that consumers can go there and get stuff. These are business-to-business suppliers.

Service provider: Places where the supply of goods is negligible with regard to the services provided. Examples: an event location, a university, a cleaning service.

PlaceDescription with PlaceDescLanguage
Now, here are two important things to note:

The PlaceDescription as well as other Descriptions throughout the project need to be of a data type that can be filled with lots of bytes of text in a markup language. The Connected Web Applications need to be able to display the rendered text in a nice way.

For all fields that can reasonable be filled in different languages there is the field for the embedded table that ends in ...Language. In this particular case you have an embedded table with two fields: PlaceDescription1...X | PlaceDescLanguage1...X. The PlaceDescLanguage field needs to be filled by the user with ISO codes from the LangCodes table.

PlaceDatetime
The Combined Web Applications need to make sure to evaluate the two PlaceDatetimeKeywords: start, end. This is useful for searching only up-to-date items.

PlaceRequiredValCat
This field is not used currently. It will be explained in the future documentation of Valuations and ValuationCategories.

PlaceLinkURL
In a future version of this documentation a new window will open where you can enter further details on the link. It will be connected to an extra table for URLs. For now, users can only enter the URLs.

3.2.4 Field category 'special'

Fields of this category only apply to the records for which this table is designed.

PlaceCoordinates
This field description needs to be updated with input of the developers of OpenFairDB.

PlaceEmail
Connected Web Applications need to make up their minds about whether they want to collect and display email addresses exposing Places to spam.

PlaceOpen fields
This embedded table is used to enter the opening times of the Place. Whether the data structure is useful or whether better ones exist, needs to be determined in discussions with the developers.

3.3 Data fields of the Effects table

This table does not repeat all the information that is valid for both the Places and Effects table. For any explanations about the meaning of technical, metadata, and common columns the section above about the Places table columns are highly recommended.

EffectLinkURL
An additional table for URLs and a corresponding form is planned for the future. As of today URLs can simply be entered here.

Until now the Effects table was very straight forward. The next fields are more interesting. The next three have to be entered in accordance:

EffectTransferPhysicalQuantity
Please enter the physical quantity in which the effect is usually transferred, eg volume, mass, energy, or – although it is not a physical quantity – number.

EffectTransferUnit
In this field, a unit that fits the EffectTransferPhysicalQuantity needs to be given, eg meter for something that is sold by length.

EffectTransferAmountNumber
Here the amount or number of the unit in which the Effect is usually transferred needs to be provided by the user. These 3 fields give us an important data point that we need for the next steps. We need this data point to build relations to other physical quantities associated with the effect:

EffectOtherTransferPhysicalQuantity
In this embedded table, users can enter other physical quantities and amounts to which the main physical quantity corresponds, eg (1 cubic meter of biogas corresponds to) 11.36 kilo-watt hours of energy, comment: “if combustion is optimal”. Another entry for biogas could be: (the volume of 1 cubic meter corresponds to the) mass of 1.394 kilogram.

This embedded table for the EffectOtherTransferPhysicalQuantity fields can be used to calculate for instance a CO_2 footprint for a product by using the component list. The component list will be explained in greater detail below. Quick example here: There might be a product which needs 100 kWh of energy and it is known that the energy source is natural gas. In the record of the Effect for natural gas in the city of production, the EffectTransferPhysicalQuantity is 'volume' with the EffectTransferUnit 'cubic meter' which does not match the component list of the downstream product. With an entry in the EffectOtherTransferPhysicalQuantity embedded table, an algorithm can use the natural gas Effect record anyway although 'energy' in 'kilo-Watt hours' is not the main physical quantity.

EffectSpec
This embedded table allows the user to enter an unlimited number of specs of a product or service.

EffectUpstream embedded table, 'component list'
These fields come almost last, but are very important. Please note: Negative upstream effects of the production of the current Effect have to be included in the component list.

EffectUpstreamNo
All components are numbered automatically.

EffectUpstreamID
An EffectID to identify each component can be provided here. The user needs to be able to search for an existing one. The field can be left empty if the component in question does not yet exist in the Effect

table.

EffectUpstreamName
If an EffectUpstreamID cannot be given, a name as a place holder can be entered here. Once an EffectUpstreamID is given, the EffectName is displayed here and the place holder is no longer shown.

EffectUpstreamTransferUnit
This is the unit in which the amount of the component is measured. It can be 'piece' for a resistor or 'pound' for flour. If an EffectUpstreamID is provided the Connected Web Application should suggest the EffectTransferUnit of the component together with the its units in EffectOtherTransferPhysicalQuantityUnits. The suggested units can be overwritten.

In case the unit provided by the user is not part of the EffectOtherTransferPhysicalQuantity list of the component, the Connected Web Application should issue a warning.

EffectUpstreamUsedAmtNo
The users enters the amount of the upstream Effect that goes into the currently entered Effect.

EffectIsCompleted
For an analysis of cumulated effects, it is very important to know whether the work on an Effect is still in progress or whether all past effects of this Effect have been included in the component list. Users are requested to set EffectIsCompleted to 'true' if the component list has been 'completed'. This is usually only possible by using estimates.

EffectIsCompletedComment
Is it ever possible to include all former effects into an Effect? Expert users are asked to comment on the lifecycle boundaries assumed for this Effect.

3.4 LangCodes table

The most special thing about the LangCodes table is that the records are not indexed by the Central Database, but by the user by giving the 639-3 language code (https://en.wikipedia.org/wiki/ISO_639-3).

In order to avoid the criticism around ISO 639-3, users will be able to enter codes that do not exist, yet. Users should try to rely on existing codes if it is acceptable to the speakers of that language. If not they can enter their own code and language or even change an existing code.

Please note the existence of the new language code 'inw' (internationalized, western style) that is supposed to be used for internationalized connotations of names in non-Latin alphabets, Combined System-wide.

All data fields in this table should be self-explanatory from the 'explanation' row.

3.5 Units table

This tables serves as a source of information on how measurements from on unit translate into another one. In this table, users again have a huge responsibility in avoiding nonsense.

UnitID
As in the LangCodes table the Units table ID is not set by the system. The user creating this record is required to use the internationally agreed symbol for that unit, ie: kg for kilogram, ft for 'imperial foot'.

UnitRecordStatus
This field can be set to supported by the administration of a Connected Web Application if the record is reliable.

3.6 Pictures table

The meaning of the fields in the Pictures table is intended to be self-explanatory.

4 User interface

The description of the user interface for working with the information from the different tables will require a document in addition to the present one. Some general considerations are given below.

4.1 Trust users

The combined system should be designed for trusting users to do the right thing. This is intended to reflect the human dignity of users and to welcome their motivation.

4.2 What the interface must be able to do

4.2.1 Search

A search page is the central start point. The system should also recommend a search to all users intending to put in a new record to avoid duplications. The search page needs a result page, of course.

The result page for places and events should contain small map views. As a result page for the places and events of a whole city, a big map view is needed. This has already been implemented in Karte von morgen.

4.2.2 Display data records

A record display page should offer a neatly presented digest of the record as well as a functionality to view the whole set of the combined 'common' and 'special' records. 'Combined' in this context means that for every field only the latest entry is displayed.

A challenge is the presentation of the many embedded tables in each record. A meaningful number of the most important items in these embedded tables should be visible.

Also, there should be an extra page that allows to review the entire history of the record including the 'technical' and 'metadata' fields.

The users should be able to restrict the data displayed to them to the 'supported' versions of the records.

4.2.3 Data entry

For data entry into the Places table, the data entry form of Karte von morgen is the point to start further enhancements, https://www.kartevonmorgen.org/.

Choice of form fields
For data entry, the data fields can be put into a more convenient order other than the order of fields in the database description table. A reduced set of form fields is possible to allow for an easier entry of basic field. An expert form should contain all the available fields of the table.

Pre-fill existing values
For existing records, when data for the record is edited or amended, the existing data needs to be pre-filled as default values in the form fields.

Keyboard navigation
It should be possible to fill a form and switch to a new form without hands leaving the keyboard, preferably using the usual keys and short-cuts.

Accept almost any value
The Connected Web Application should accept any values in most field, except where the functioning of the database is threatened. This is also important to avoid interlocking requirements. A user might have to enter a Unit that does not exist, but cannot finish the Effect entry before the unit is entered.

Embedded tables
A convenient method needs to be created to fill the embedded tables.

Filling of existing IDs
Wherever an existing ID from a table is required, a search functionality for that ID needs to pop up, eg where the IDs of Effects that can be bought at a Place can be entered in that Place's record. For every field ID that shows up in the table template row 'Reference to master field in other table' there needs to be this functionality.

Iterations
If the search for a value from another database table produces no results, the user should have the opportunity to create that record in the other table. For example, if a component list cannot be completed because a component is missing, it should be possible to open a second instance of the Effect data entry page in order to enter the missing component.

This iteration is only possible for one step in order to avoid confusion. The second Effect would have to be entered in a way that cumulates all its upstream components. To improve it, the user should open it again later from the data entry page.

Copy records

It should be possible to copy records, make changes and save them under a new ID.

List of incomplete records
A functionality to identify and find incomplete records would be a useful feature for management of every user's own entries and also for people who want to enhance available entries of other users. What will be the indicators for 'incomplete records' will be the object of some pending research.

4.2.4 Necessary Research

Component list
The ability to link an Effect to its upstream Effects is potentially a powerful source of insight. The question is still how good it works. What is the quality of the results produced?

The data gathered with the Combined System is open to researchers. Interesting question are: What are the challenges in calculating the total footprint of an Effect from the component list? How to display existing (and lacking) data in the supply chain that led into an Effect?

Loops
How do we treat loops? For example, when produced wheat grains contain the same kind of grains as a component for its production. (In former times, this was the rule, not the exception.) One of the systems needs to stop the regression at some point.

5 Comments on data entry

Different information requirements for one product
If users or groups of users have different requirements regarding the quality available in the record, they need to create two different records. Example: For some purposes it might be sufficient to include a rough guess of cumulated CO_2 in the component list for a football. Here an Effect record with EffectIsCompleted = 'true' is recommended. While in other cases of ongoing research, it is better to work with a copy of the Effect that does not rely on estimates.

When creating a an effect that is 'completed', estimates for missing effects need to be included in the component list.

6 Concluding remarks

As readers might notice: This text is work in progress. They are asked to feel free to address remarks and questions to the author via Github or socialmedia (att) ourconomy.com .