User Interface and Data Visualization for Environmental Assessment

Jonya Chen

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2017-43
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-43.html

May 10, 2017

Acknowledgement

Special thanks to my teammate, Caroline Chen, our PhD advisor, Jeremy Faludi, and our faculty advisor, Prof. Bjoern Hartmann. Also thanks to the Berkeley MEng EECS program for giving us this opportunity.

Final Capstone Report

User Interface & Data Visualization for Environmental Assessment

Jonya Chen

May 10, 2017

Executive Summary

Life Cycle Assessment (LCA) is a powerful method in the field of sustainable design used to quantify the environmental impacts of a product or service. It allows users to input various materials or processes and outputs impacts such as greenhouse gas emissions, pollution, toxins, and more. Not only is LCA beneficial for telling the backstory of a product, but it also helps to identify the biggest environmental impacts. It guides the process for product development and helps customers establish a baseline for improvement to make more informed decisions. However, as of now, few companies use LCA because the currently existing software user interfaces are confusing and hard to follow. Furthermore, the level of experience required to perform the analysis on the current tools is very high, stemming from a steep learning curve and outdated software interfaces. In an effort to expand the LCA user base, specifically towards product designers interested in analyzing the sustainability trade-offs between various materials, our capstone project focuses on redesigning and implementing a new LCA user interface. Through an iterative process of user studies and design prototyping, we hope to allow users to build models and create better data visualizations that are both engaging and informative. Ultimately, LCA software adoption can greatly increase industry's ability to produce greener produce and reduce systematic impacts.

Throughout the year, my team worked to design, prototype, user test, and implement an LCA interface that is functional and user-friendly. In the end, we were ultimately able to accomplish our goal of having working software that is able to collect user input for a product and output useful information about that product's environmental impacts. Chapter 1 of the report will detail our technical contributions, including how we created our interface prototypes and developed them to get a final product. Chapter 2 of the report will review our project's context, as well as show the significance of LCA software in general.

Chapter 1: Technical Contributions

1.1 Introduction

The purpose of this chapter is to detail and analyze the technical contributions accomplished throughout the year, as well as identify encountered successes and challenges. For this part of the paper, I will take a look at the work I have done with respect to design, prototyping, and code implementation. Overall, I hope to detail our project progress, as well as showcase a successful year that resulted in a completed, valuable life cycle assessment product.

1.2 Work Breakdown

Our team decided to break down the overall project work into two main stages: prototyping and implementation. We spent the fall semester working on finalizing a prototype for the user interface and data visualization, while the spring semester was spent implementing the actual software platform. For the fall semester, my teammate, Caroline Chen, and I collaborated the entire time on brainstorming design ideas, prototyping them, and through user testing, filtering out which features worked better than others. We continued the same process over and over again, until we had a finalized design for the interface at the end of the fall semester. Then, in the spring semester, we took a divide-and-conquer approach with the technical implementation. While I focused more on coding the front-end user interface and model building, Caroline worked more on implementing the back-end infrastructure, databases, and data visualization. While we worked more independently throughout the spring semester, we made sure to constantly keep each other updated in order to stay on the same page. By the end of the year, we then combined our work in order to have a final working software interface for LCA.

For the following sections of the paper, I will first details our progress during the fall semester in the two main design stages, prototyping and user testing, as well as analyze any accomplishments or challenges involved. I will then discuss specific work that I did during the second semester which involved implementing the actual user interface and the development tools I used to do so.

1.3 Sketching and Prototyping

The first step of our project was to gather inspiration from previously existing LCA software. We explored different software including GaBi[1], Sustainable Minds[2], openLCA[3], and SimaPro[4]. We analyzed various interfaces and received feedback from experienced users on areas of improvement, which gave us a sense on how to differentiate our own product. The main flaws in the existing interfaces we saw stemmed from a chaotic workspace and workflow (Figure 1 in Appendix). In addition, most of the software had very complex data visualization (Figure 2 in Appendix). Overall, there are various design usability heuristics that are violated, including a lack of visibility of system status, flexibility or efficiency of use, and minimalism design. These weaknesses resulted not only in user confusion, but also in a higher learning curve required for user understanding. Therefore, we decided to create our own LCA interface that would allow users to build models more intuitively and create better data visualizations that are engaging and informative.

We then went to the drawing board and brainstormed a few ideas for the general user flow of the software interface. Since our target group included users without previous LCA knowledge, we had to carefully make sure that each of our designs were intuitive and easy to follow, regardless of the user's previous knowledge or background. Users should be able to use the software in a quick and efficient manner, such that there would be much less confusion when it came to building models. As a result, we decided to create low-fidelity prototypes, which allowed us to quickly test simple concepts through raw and limited functionality prototypes. This technique would also ensure that users were not biased upon tiny details throughout the testing process.

Through our prototyping, we considered various design decisions, such as having the interface on one-screen versus multiple screens, using tabs versus drop-down menus, including pop-up

[1] "GaBi Software: Life Cycle Assessment LCA Software." http://www.gabi-software.com/. Accessed 10 May. 2017.
[2] "Sustainable Minds." http://www.sustainableminds.com/. Accessed 10 May. 2017.
[3] "Open LCA." http://www.openlca.org/. Accessed 10 May. 2017.
[4] "SimaPro - PRé Sustainability." https://www.pre-sustainability.com/simapro. Accessed 10 May. 2017.

screens, and much more. As such, we added a subset of each feature group to our prototypes, which were first on plain paper (Figure 3 in Appendix). We then created low-fidelity interactive prototypes through the use of notecards, where each screen or interaction would be represented by a new card (Figure 4 in Appendix). We also used the “Wizard of Oz” design methodology when testing the notecards, which allowed us to fake various behaviors of the product by manually switching between cards instead of automatically through the use of a computer. According to a paper by Lennart Molin, this prototyping technique can also promote a more collaborative dialogue between the user and the tester, which can then further strengthen the user as a stakeholder (Molin, 2004).

Our final phase of prototyping involved computer-generated prototypes that users would be able to interact with by clicking on an interface on a monitor in order to see feedback on the interface. We used Microsoft PowerPoint in order to create these slide decks for various user interface designs and presented them to our users (Figure 5 in Appendix). We were then able to gain much more insight into which interactions were effective and clear to understand versus confusion, mainly because interaction feedback was now easier to visualize.

Another key part of the LCA process in our software was the data visualization, which is what a user sees once he/she is done building the model. So we then conducted the same prototyping process described above with respect to the data visualization aspect of LCA. We first created various designs involving comparison of the environmental impact of an entire scenario (Figure 6 in Appendix). We also created models that would vary the way a user viewed the data, whether it was stacked vs unstacked bar graphs or sorted vs unsorted bar graphs (Figure 7 in Appendix). An additional functionality we wanted to incorporate with our data visualization portion of the interface was also to allow users to be able to “zoom-in” to an environmental impact analysis and determine which specific materials or procedures were causing the most impact. Users would then more easily be able to identify which parts of their model to modify or substitute in order to have a more efficient sustainability trade-off. Therefore, we also created a few prototypes of this analysis workflow in order to test out which version prompted the most intuitive user interaction.

Finally, one of the main differentiating factors we wanted to incorporate to our version of an LCA software interface was the addition of uncertainty in the impact analysis. Since we noticed that not many currently existing LCA interfaces allow for a way to quantify uncertainty and its effect on end results, we wanted to include this beneficial feature into our own interface. The addition of uncertainty analysis could help users save a lot of time since they will no longer have to spent time and effort to collect unnecessary data. As a result, we also prototyped a few different designs for how a user can view this uncertainty, including various graph formats such as violin plots, box plots, and error bars (Figure 8 in Appendix). Overall, through our designed prototypes, we were able to not only create variations of how a user might interact with our interface, but also utilize them in user testing.

1.4 User Testing and Analysis of Results

One of the most important steps of our work throughout the fall semester was user testing. The process involved presenting users with our prototype, observing as they interacted with them, and analyzing which features were best suited for each interaction. While it was a time-consuming task and required at least 5-10 hours on testing weeks, it was a very crucial accomplishment to our progress because the testing results helped us decide which design changes to make for the next prototyping iteration.

A major road bump in our project occurred during one of our first rounds of user testing. We noticed that we were starting to get a variety of opinions from our users - there didn't seem to be a clear enough general consensus on which features were superior to others. Our team eventually realized that the source of the problem came from us initially having a wide set of user testers, which consisted of both expert and non-expert LCA users. As such, the testing result variation became a challenge for us because it became very difficult to narrow down our design from current prototypes, like we had originally planned. For example, we found that our expert LCA user testers preferred to have software with additional functionality, such as tools to help determine the source of materials within a bill of materials or having a set panel where all processes are viewable in one space. In contrast, our engineering users preferred a layout similar to that of Autodesk or Eclipse software, which has a one-screen "homepage" for the entirety of the product's use, and our non-expert LCA users preferred to have a chronological tab list to organize and walk them through each step of the LCA process in the software. Essentially, we realized that experts and non-experts have a very different method of performing LCA and our user interface would not be able to account for both since the user testing feedback was too varied. In the end, we decided to narrow down our target user group in order to design specifically for non-expert engineers, which helped us to create our future interactive prototypes. We were then able to gather a consistent set of users (mechanical engineerings in the UC Berkeley MEng program) who we knew would fit our vision and be able to give us related feedback on our designs. As a result, we could then move forward with building the minimum viable product for a unique user group.

After several cycles of iterative prototyping and user testing, we were able to gain a lot of valuable information about user mental models when performing LCA. At that point, we were able to understand how users wanted to build their assemblies and interact with the interface. We then took the feedback and used it to refine our prototypes. By the end of the fall semester, we were able to have a final prototype of both the user interface and data visualization for our LCA software (Figure 9 in Appendix). As can be seen, users liked an interface where they could drag and drop the blocks onto an overall screen. The library of materials and processes would be consistently visible throughout the entire process. In addition, users would also be able to modify properties of the materials or processes on screen by just clicking to edit. Once we had this final interface design, we were then ready for the next phase of our project: implementing the software.

1.5 Technical Implementation

In order to get started on coding during the spring semester, I first needed to start with a code base. Fortunately, during the previous semester, our team had already been collaboratively working with a group in Berkeley's CS 169: Software Engineering course. They helped us set up an initial web application that was deployed on Heroku, a cloud provider service. The web application had a basic user interface set up with some model building components implemented similar to those in our initial prototypes. While their work only spanned a few months during the fall semester, my goal during the spring semester was to continue working on that initial code base to implement further key features of our LCA software.

In terms of technical tools, languages, and platforms used for development, I decided to use HTML5, CSS, JavaScript, and jQuery for the client side and Rail 4, Ruby 2 for the back-end, server side platform. All code was open source and stored in a repository hosted on GitHub.

My first focus was to modify the existing CS169 code base to something more similar to our final user interface prototype for model building. I altered some of the styling such as how a user dragged and dropped to user model panels or over existing materials/procedures. I also coded in additional life cycle categories. These categories would allow the user to specify the lifespan of a product, frequency of its use, as well as what happens to the product after it has been used (i.e. landfill or recycled). Furthermore, I also modified some of the functionality of the model building that had restrictions on which materials or processes were draggable and droppable. I removed some of the constraints so that all materials, including material subcategories, could be a standalone component on the model building page. In addition, procedures were no longer required to only apply to a material.

Initially the code base was also set to work with a dummy database. However, in our case, our final interface needed to work with actual data from Ecoinvent 3.2[5] database files. Ecoinvent databases provide us with up-to-date and well documented information about process data for thousand of products. Using this information, we would then be able to run calculations and graph environmental impact in the data visualization part of our software. As previously mentioned, most of my teammate Caroline's technical work involved setting up this back-end side that would then be used to display the results from after the model building. Therefore, one of my main tasks was to be able to set up the model building portion to sync with her back-end infrastructure. This involved two main components: syncing the library display and being able save model building information to pass onto Caroline's data visualization code.

In order to sync the library display, I would need to display all materials and procedures using a specified set of table definitions used by the database (Figure 10 in Appendix). I implemented this by using Ruby's ActiveRecord Query Interface[6] in order to find specific database records to display in the model building page. To retrieve specific objects, I would using the .where() method with conditions on both parent_type and parent_id. The retrieved objects would then all be within the same category and and the same type of parent. For example, if I wanted to retrieve all subcategory headings of materials in my database, I would create a query that would gather all objects with parent_type "Category" and parent_id of "1". In this case, a category is just each stage of the LCA (material, process, usage, transport, end of life) and the id of material is 1. I then created similar customized queries for all other part of the library display and passed the

---

[5] "ecoinvent 3.2 – ecoinvent." http://www.ecoinvent.org/database/ecoinvent-32/ecoinvent-32.html. Accessed 10 May. 2017.

[6] "Active Record Query Interface — Ruby on Rails Guides." http://guides.rubyonrails.org/active_record_querying.html. Accessed 10 May. 2017.

retrieved information to display using Materialize[7], which is a responsive, front-end framework implemented with CSS and JavaScript.

Once our model building libraries was synced, the user would then be able to drag and drop various components to their overall model. The next step was to pass the model building information to Caroline's back-end code, which would then run queries over whichever items were chosen during model building in order to get the respective environmental impacts. There would then be various calculations run from the quantities and units using a specified LCA method, which would then result in graphs. In order to save model building information, such as the activity id, quantity, and units used, I created a hash function, edited it to the specific format that we wanted, and then stored all the information in a JSON file. This JSON file is then read by the back-end code, which will then take the user's model building information and display environmental impact results for that particular model.

Finally, the last parts of our technical implementation involved adding our own new features to the interface to make it more user-friendly. This included tasks such as allowing users to have drop-down options for units, adding a search bar functionality within the model building library to make it easier for users to find specific materials/procedures, and accounting for various formatting features (naming conventions, coloring, block sizing, etc). I also implemented a function on the user interface where users could create subassemblies within their model and further streamline their building process. Lastly, I made sure to account for edge use cases which might cause more user confusion, such as accidental dragging of library headings. In the end, we were able to successfully accomplish our goal of having a completed, user-friendly LCA software interface.

1.6 Future Work

While our team was able to implement a final LCA software interface, there are several additional features that could be implemented in order to take our product to more than just a minimum viable product. Currently, our software only accounts for one methodology of LCA analysis, ReCiPe[8]. ReCiPe specifically determines indicators at two levels; midpoint and endpoint indicators, along with normalization. In order to increase the chances of having our software adopted by more LCA users, we would need to allow for more ways to interpret the impact data, such as TRACI, CML, and ILCD/PEF. In addition, while our initial goal was to be able to connect our software interface to an actual LCA database, we ended up having some difficulty in the end due to the complexity of the existing database that prevented us from migrating it over to our Ruby on Rails application. Therefore, a major future step for improvement would be to figure out a framework that would allow us to fully integrate imported databases into our software. Finally, additional user testing could always be done to continue on the constant iterative cycle to further improve our product.

[7] "Materialize." http://materializecss.com/. Accessed 10 May. 2017.
[8] "ReCiPe." http://www.lcia-recipe.net/. Accessed 10 May. 2017.

1.7 Conclusion

Overall, we made a substantial amount of progress this year in terms of implementation. As a team, we were able to create a detailed and fully-implemented user interface to properly perform LCA to quantify the environmental impacts of products. We spent time prototyping designs at various levels of detail, ensuring that each feature was easily understandable to a user and would provide the right interaction feedback. In addition, we were also very keen on bringing our prototypes to real users, who would then be able to provide us with important feedback that would keep our design cycle going. While the technical implementation was the most challenging part of the project, since I personally had no previous experience with Ruby on Rails, it was definitely a great learning experience and very rewarding once I was able to get certain functionalities implemented. Overall, the accomplishments we achieved throughout the year were very impactful to the success of our project.

Chapter 2: Project Context

2.1 Introduction

The purpose of this chapter is to take an in-depth look into the potential of our LCA software in real-world markets and more specifically, the packaging industry. This paper is part of a group effort to analyze position and impact of environmental assessment software in industries. My partner, Caroline Chen, takes a look in her paper at how a growing interest in building certification and credit programs create an incentive for more companies to adopt LCA practices. In particular, various enterprises that have programs like environmental management system (EMS) have seen an increase in successful sustainability results stemming from their use of LCA. As a result, more companies and industries can follow a similar suite and benefit from these tools. To complement that idea, my paper focuses on how another industry, the packaging industry, can adopt similar LCA processes, such as our software, in order to create "greener" products as well. By analyzing the packing industry's current motivations for maintaining environmentally friendly practices, we can determine the best strategies and methods to position our LCA software in order to provide the most value in that particular market.

2.2 The Food Packaging Market

The packaging industry has generally been stable. As consumer spending continues to grow, so will demand for packaging services. A major example of a powerful segment in this industry is food packaging. According to food analyst Sarah Levesque, food companies "need to find innovative ways to cater to demand for convenient, transparent and environmentally responsible packaging while providing consumers with the utmost value" (Levesque, 2013). Nowadays, consumers are demanding more from their food products since there are so many options to choose from. As a result, brands resort to food packaging in order to meet these consumer needs. The food packaging industry is major - the creation of new food packaging has increased 89.3% within only a five year span between 2009 - 2014 (Topper, 2014). Moreover, the percentage of new products has declined 37.5% (Topper, 2014), indicating that companies are now deciding to repackage existing products to attract new customer, rather than focus on creating new foods in general. Furthermore, the competition in this industry is high, as a majority of packaging industry operators are small businesses (Robert, 2016). There is also low capital investment necessary for companies to enter the market, resulting in a low level of concentration, with "the top companies estimated to account for less than 14% of industry revenue in 2016" (Robert, 2016). As a result, competition is high within the industry. Thus, food packaging companies are constantly looking for ways to position themselves in the already popular and competitive market.

2.3 Sustainable Packaging

One key way for food packaging companies to differentiate their product is to offer more sustainable packaging options. According to recent marketing research, more than 53% of consumers say that packaging indicating environmental or sustainability efforts is important to them (Topper, 2014). In fact, they believe that the more environmentally friendly a product is, the more likely they're influenced to buy that product over another (Topper, 2014). Allowing users to buy more sustainably packaged products helps them feel more comfortable with the quality of the food itself. People care about being green - in fact, a majority of the population (73%) believe that each of us has an obligation to be environmentally responsible (Levesque, 2013). Companies that strive to be sustainable can not only stay ethical, but also attract customers to benefit business. This idea is backed by the fact that 61% of consumers believe companies who strive to be environmentally friendly in their practices will have a better business overall (Levesque, 2013). Furthermore, green packaging is very important amongst high-income customers. A 2013 marketing study showed that customers who earn more than $150K are more likely to purchase a food product that has environmentally friendly packaging (Levesque, 2013). Therefore, companies should consider pursuing more environmentally friendly packaging in order to generate a higher value of quality and maintain their business from higher-income consumers.

2.4 Valuing the Technology

How do these companies in the food packaging market find ways to create more environmentally friendly packaging? One powerful technique is through the process of Life Cycle Assessment software (LCA), which was mentioned earlier in the introduction of this paper. By performing a quantitative evaluation of a food package's environment performance across its life cycle, companies can become more sustainably aware by making different packaging design choices in order to reduce environmental impacts. This same idea can also help the companies further differentiate their products versus other competitors in the market. Upon researching which tools currently exist for analyzing sustainable packaging, the main one that is commonly used is called PIQET[9] (Packaging Quick Impact Evaluation Tool) (Verghese, Horne, & Carre, 2010). The tool is unique in that it streamlines the LCA process so that results can be quickly analyzed and decisions can be reached, all without the usual complexity and detail. If we were to commercialize our capstone technology, we could strategically position our product so that it's attractive to other markets in a way that's similar to currently existing options for the food packaging market, such as PIQET. However, our interface will be a more updated and

[9] "PIQET." http://piqet.com/. Accessed 10 May. 2017.

user-friendly version, allowing us to market ourselves as a tool that can be utilized even without prior LCA knowledge. The utilized market strategy involves using market segmentation to target smaller groups of users that could benefit from our product directly and advance them to leaders within their own industries. As demonstrated previously, LCA tools could be very useful for those who need design decision support when producing sustainable packaging - this similar benefit could also apply to other markets and companies.

2.5 Conclusion

Overall, environmental assessment tools and specifically, LCA, proves to be very useful for markets such as the food packaging industry. The practices provide value to companies by not only helping them differentiate themselves with respect to other companies in the packaging industry, but also providing them with powerful information to create the most sustainable packaging. This idea holds true especially since it has been shown that consumers care about what values are upheld during the manufacturing of food packaging and are more or less inclined to buy a food product based on its sustainable packaging. As a result, the use of LCA tools through the process of creating more eco-friendly food packaging could indicate further success in other similar markets.

Executive Conclusion

Looking back at the year, my teammate and I have been able to accomplish a lot. We started from having barely any clue what LCA was to ending the year with a completed software interface built from scratch. We were able to use various prototyping and user testing techniques in order to identify key features and details that designers enjoyed using when performing LCA. We then developed the interface using Ruby on Rails to bring the prototypes to life. Our final product lets users input various materials and processes, resulting in graphs that show environmental impacts such as gas emissions, pollution, toxins, and more. Our software in particular also hones in on the needs of product designers specifically, such that there is less experience required and a more engaging user flow. We hope that future users of our LCA software will find the interface user-friendly and engaging, ultimately resulting in a higher ability to build greener products in the future.

Acknowledgements

I would like to give special thanks to our PhD advisor, Jeremy Faludi. He was the inspiration behind this capstone project idea and an immense help throughout the whole process. He saw this project from beginning to end with us, and we are grateful to have been given this opportunity to work with him. I would also like to thank my partner, Caroline Chen, for sticking with me throughout this whole project. It was a great experience getting to work with you! Finally, I would to thank our faculty advisor, Prof. Bjoern Hartmann, for giving us this opportunity to work on this project for our MEng program.

Sources

Levesque, S. D. (2013). Food packaging trends - US - July 2013 - market research report - Executive Summary. Retrieved October 17, 2016, from Mintel, http://academic.mintel.com/display/637680/

Molin, L. (2004, October 23). Wizard-of-Oz prototyping for co-operative interaction design of graphical user interfaces. Retrieved April 04, 2017, from http://dl.acm.org/citation.cfm?id=1028086

Robert, M. (2016, June). Industry Report 56191: Packaging & Labeling Services in the US - Competitve Landscape. Retrieved October 17, 2016, from IBISWorld, http://clients1.ibisworld.com/reports/us/industry/default.aspx?entid=1501

Topper, A. (2014). Food packaging trends - US - July 2014 - market research report. Retrieved October 17, 2016, from Mintel, http://academic.mintel.com/display/680530/

Verghese, K. L., Horne, R., & Carre, A. (2010). PIQET: The design and development of an online ’streamlined’ LCA tool for sustainable packaging design decision support. The International Journal of Life Cycle Assessment, 15(6), 608–620. doi:10.1007/s11367-010-0193-2

Appendix

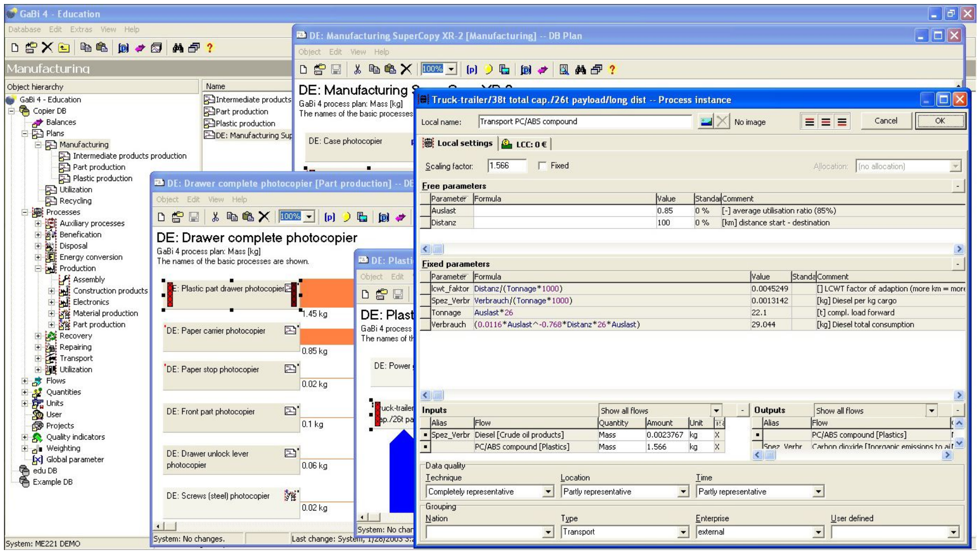

Figure 1: Currently existing LCA software with cluttered and chaotic workflow

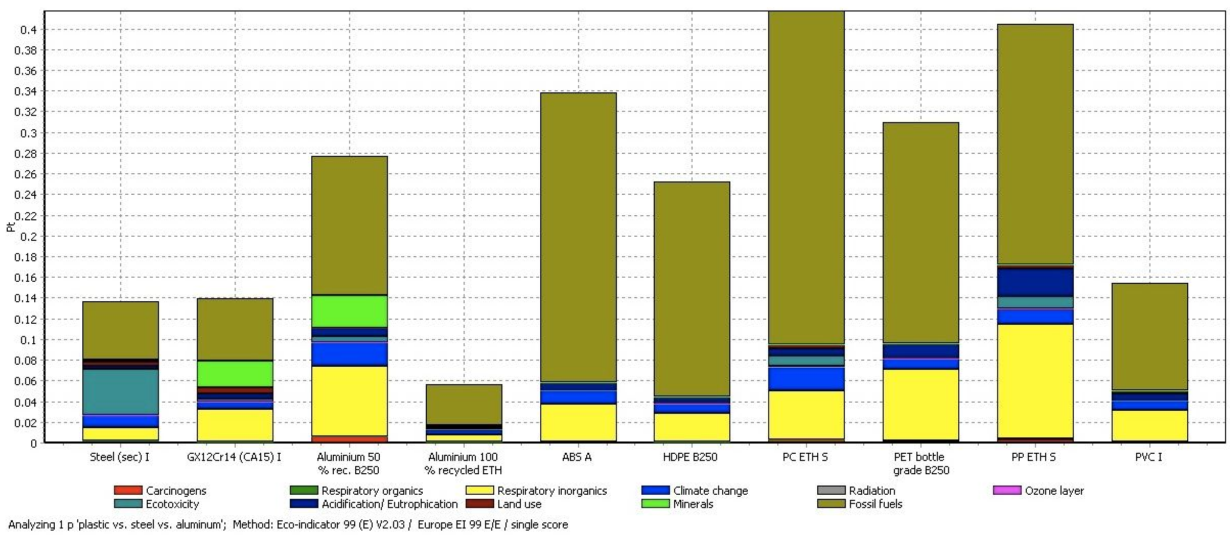

Figure 2: Currently existing LCA software with confusing data visualization

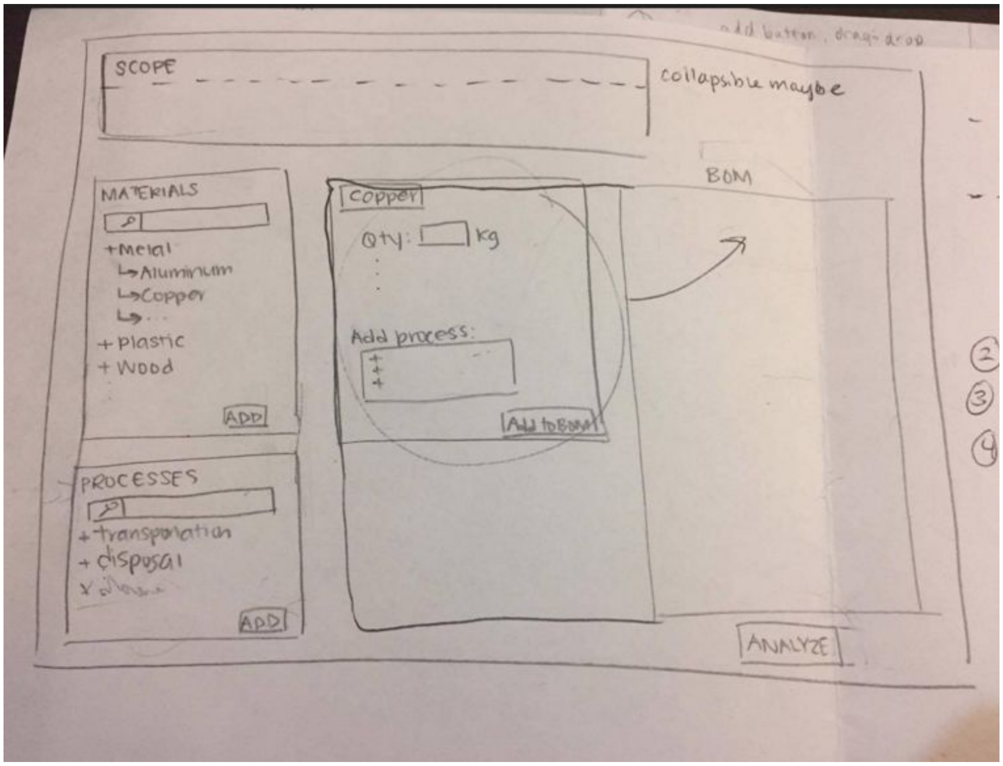

Figure 3: Example of early sketching prototype on paper

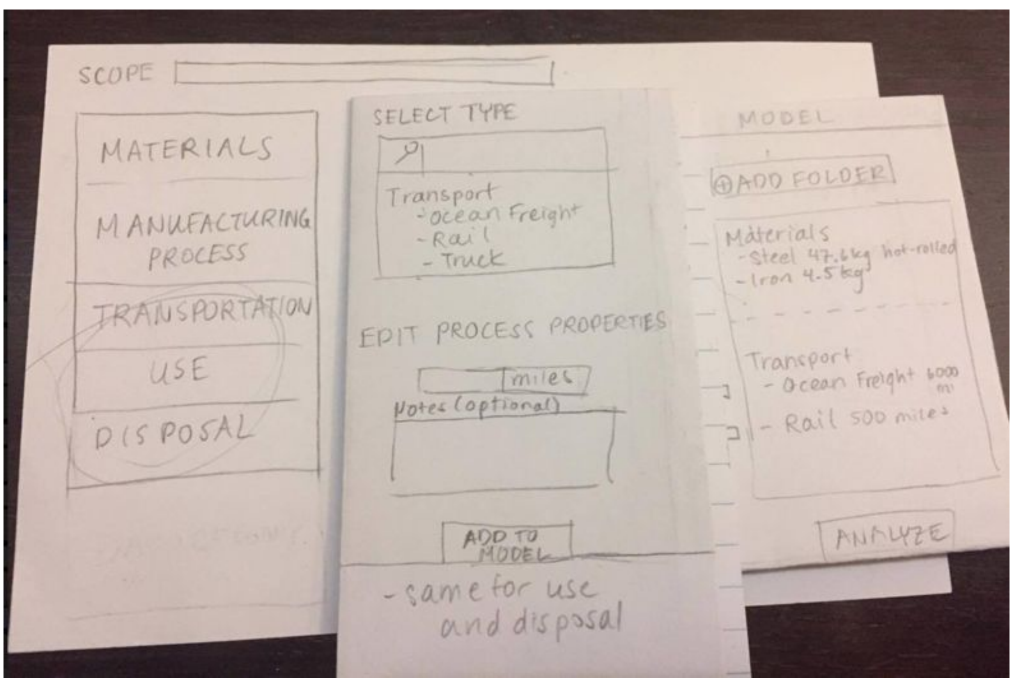

Figure 4: Example of prototyping on notecards

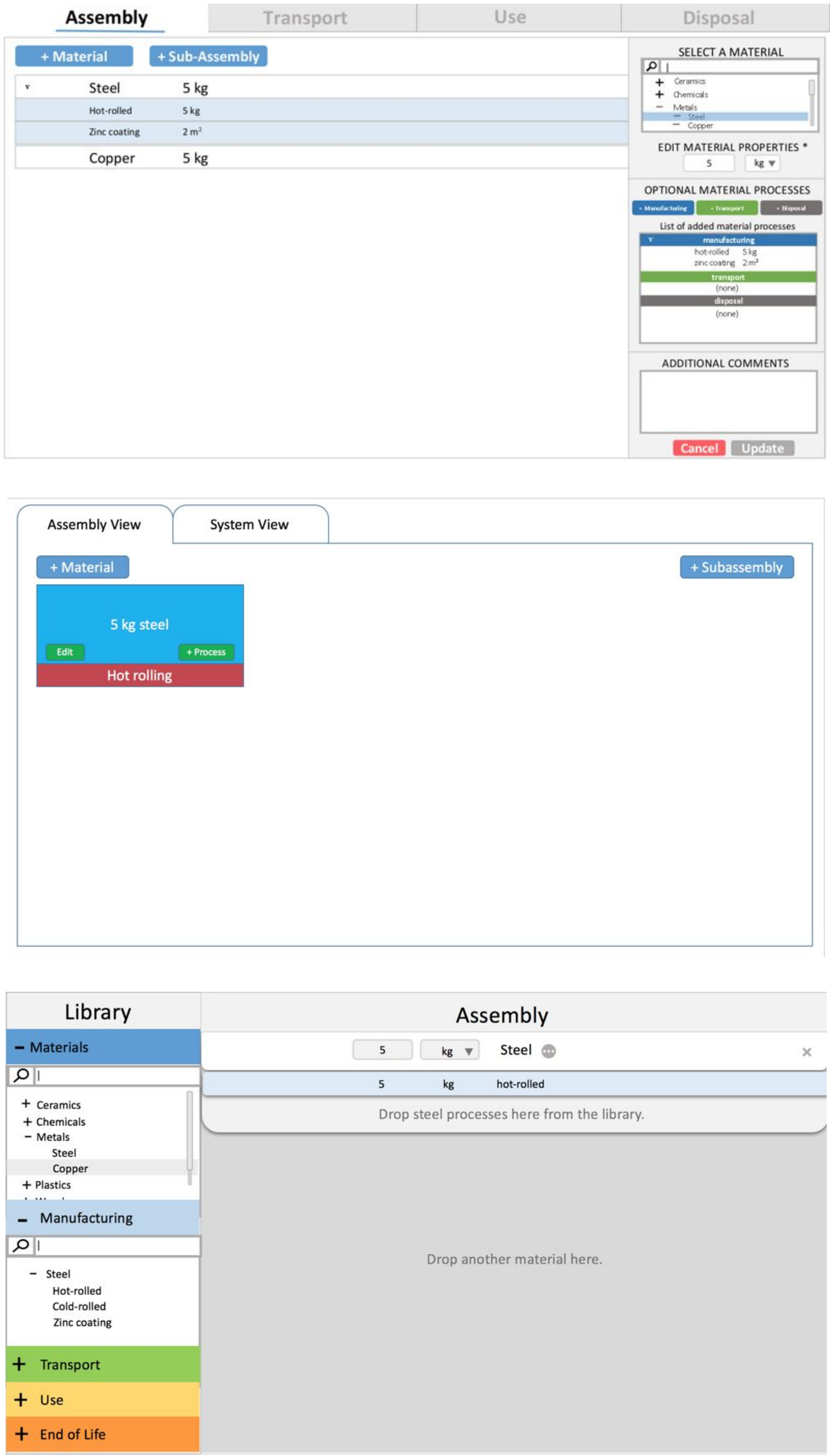

Figure 5: Examples of interactive prototypes using PowerPoint

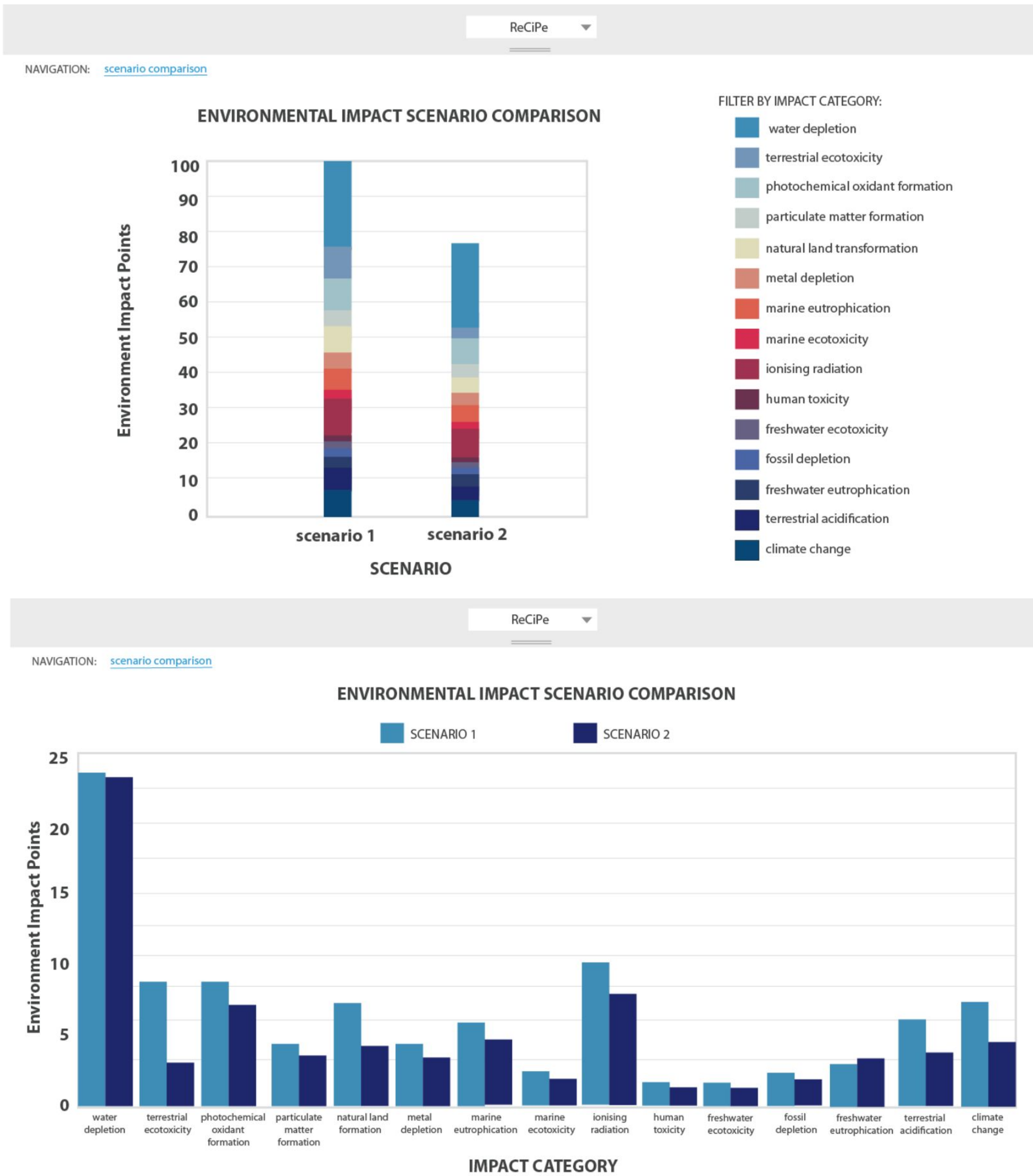

Figure 6: Examples of data visualization prototypes for overall scenario impact

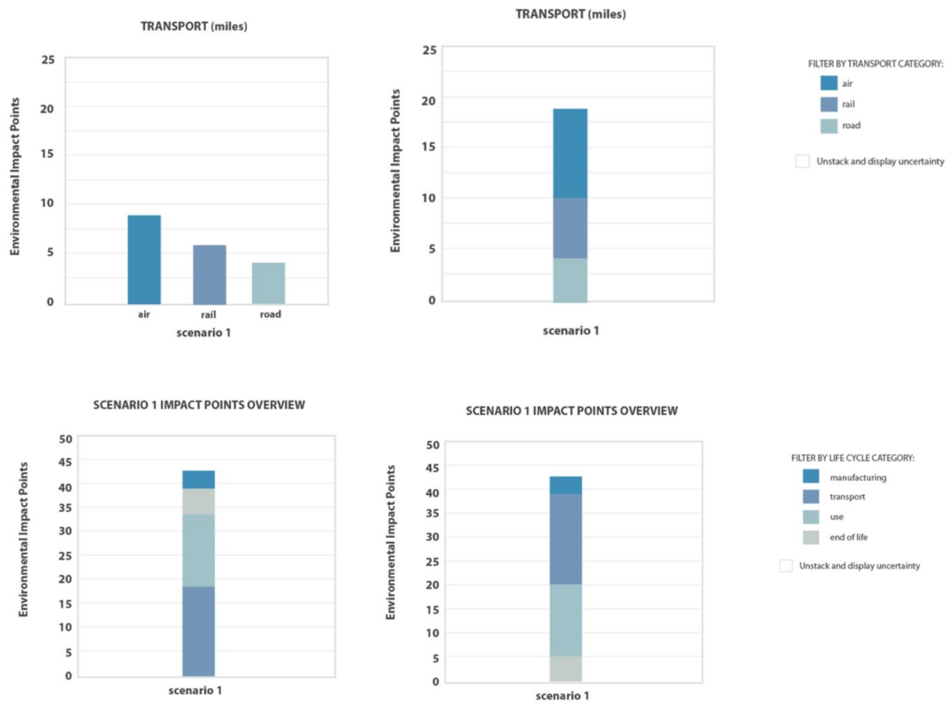

Figure 7: Examples of various formatting for data visualization

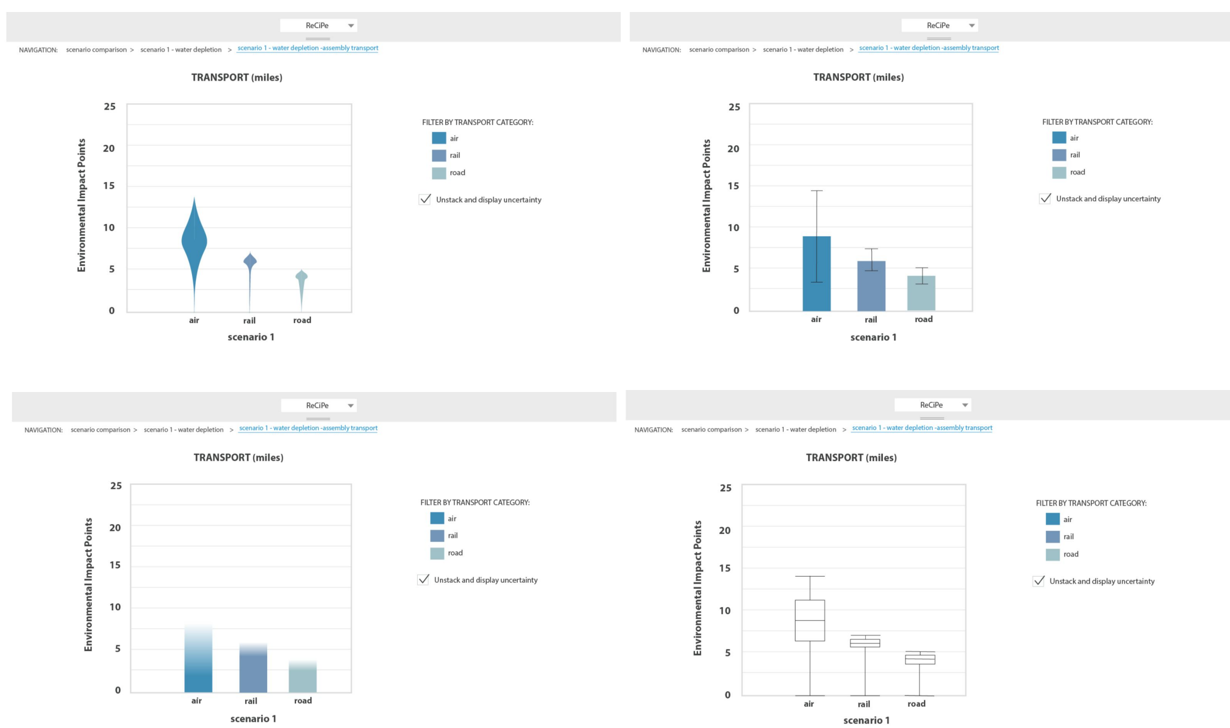

Figure 8: Examples of various uncertainty analysis visualizations

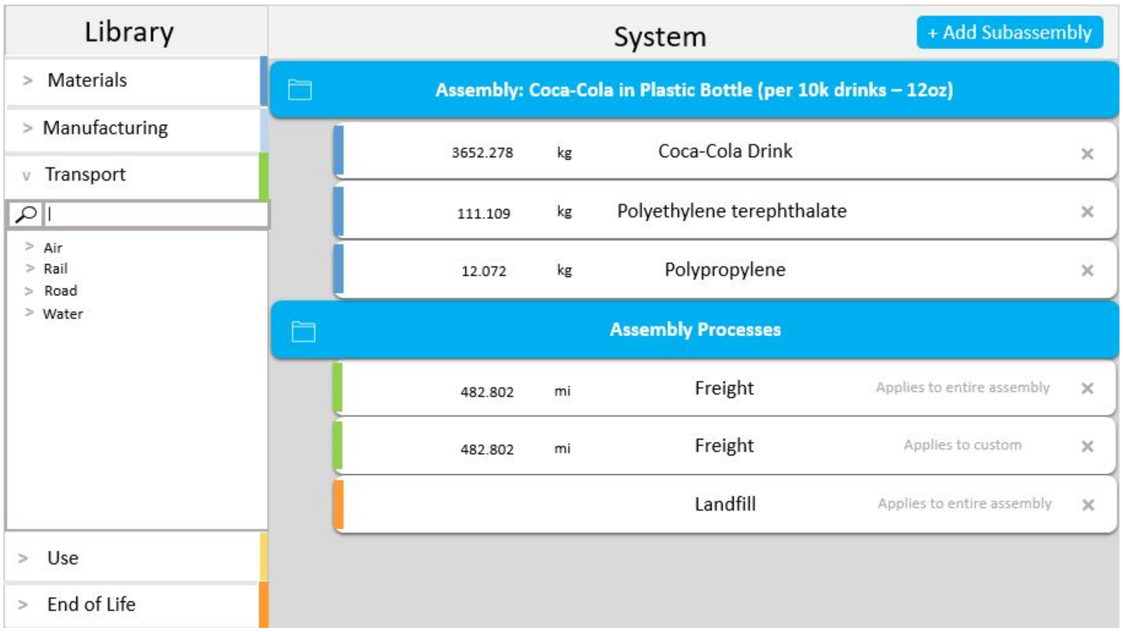

Figure 9: Final prototype of user interface and data visualization

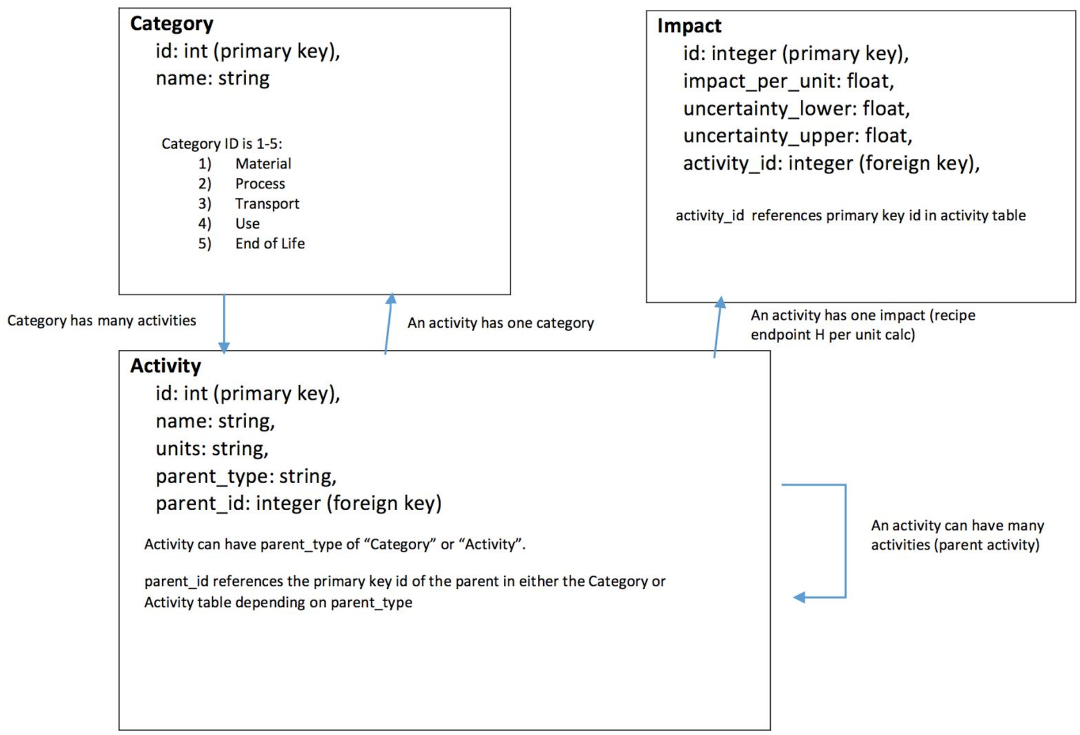

Figure 10: Database table definitions