Systems re-design for sustainability: PetShop Case study

Nguyet Nguyena
Department of Computer Science
University of Leicester, Leicester, UK

Ruzanna Chitchyan
Department of Computer Science
University of Bristol, Bristol, UK

***Abstract*—Recognition of the critical role of software in promoting and maintaining sustainability has been emerging recently. Yet, over the last 60 years a vast amount of software has been developed and deployed without any consideration of sustainability. This paper presents a case study on how relevant sustainability requirements have been identified (using goal-oriented techniques) and integrated into an existing software design and implementation. It also discusses metrics used for evaluation of the sustainability effects of the new design. The evaluation of this study demonstrates that the re-designed software is both more extendable and more sustainable. Thus, this paper presents an example process and case study for undertaking sustainability-centred software re-design, as well as reports on a number of trade-offs and interdependencies between the studied sustainability requirements.**

I. Introduction

While integration of ICT into business models and processes has a well-documented economic benefit, the environmental and social aspects of its' impact are less clear-cut. For instance, while some e-businesses, such as eBay, encourage reuse of physical goods, they can also cause over consumption [1] as well as bankrupt local producers and retailers, in turn dulling out the vibrancy of local cultures. Hence, it is evident that, in order to support long-term sustainability, such sustainability requirements must become an integral part of each software system. However, software engineers have still to understand what sustainability requirements are and how would they impact the software development process and product.

In this paper we present a case study of such requirements and the effects they have on an e-shop business for selling pets (called PetShop). The aim of this study is in determining how retrofitting sustainability into an existent system will affect that system. Following [2] we consider an activity sustainable if it's "direct and indirect negative impacts on economy, society, human beings, and environment ... are minimal and/or which has a positive effect ..." Through the PetShop case study we explore what minimising such impact will imply for the requirements, design, and operation of the resultant software. The findings from this case study demonstrate that there is an ever present tension between the three "pillars" of sustainability (i.e., its environmental, economic, and social) as they are perceived today. This is because economic sustainability for a business supported via software aims to increase consumption, while environmental sustainability aims to reduce it. The social sustainability, in turn, is in conflict with itself - aiming at both increased, faster, and more convenient consumption available to all and vibrant, un-spoiled natural environment at the same time.

We also present a process for identification of sustainability-related requirements and evaluation methods and metrics for some aspects of the economic and environmental sustainability requirements identified in this case study. Throughout this study we use the relevant consolidated research on sustainability undertaken so far in the software engineering community and extend it where open issues are identified. In particular (similar to [3] and [4], we use goal-oriented technique during requirement engineering to identify relevant sustainability requirements. The effect of the newly integrated requirements on the sustainability of the system is then measured using CO_2 emissions [5] (as outlined in III-B).

Since the present study aims at consolidating the relevant research of software engineering for sustainability, we must start by reviewing this research:

A. Requirements Engineering for Sustainability

Naumann et al. [6] propose GREENSOFT - a generic process model to study the possible causes, effects, and processes in achieving sustainability goals. This model mandates a strict relationship between sustainability tasks and software lifecycle, stating the need to observe first, second, and third-order effects in software development and use. For example, in software engineering, the 1^{st} order environmental impacts can result from electricity consumed for software development. This can be monitored via power consumption of devices. On the other hand, working conditions impact more indirectly on social values that could be evaluated via interviews and be measured e.g., via employee satisfaction levels. This work provides an overall framework on how sustainability tasks could relate to software engineering process, but currently lack the necessary substantial body of empirical evidence to exemplify its use in practice.

Cabot et al. [3] illustrate how sustainability requirements are integrated in the early stages of development through Goal-Oriented techniques in order to consider the impact of alternative decisions about conference organisation on the eco-footprint of the event (e.g., printed vs. electronic proceedings). They aim at development of a generic sustainability taxonomy without focusing on quantifying the requirements impact (except via help/hurt contributions within the goal model).

The Yellow project [4] presents a mix of requirements engineering techniques for elicitation and analysis of sustainability requirements for an event management system. This work emphasises the effect of stakeholders' (i.e., board members, project managers, technical managers and environmental champions, etc.) perspectives on both functional and non-functional requirements and proposes a process to produce a complete goal model with detailed requirements. It also considers the positive and negative inter-dependencies between sustainability requirements for the given system, but does not propagate the study to design, implementation, or evaluation.

B. Sustainability and e-Business Domain

We consider [7] a particularly relevant work to our study as it focuses on the specificities of green e-business, for which our selected case study is an example. Here a number of key factors for a green e-business are pinpointed, including:

- "Inform and Promote" through which e-business takes advantages of saving cost and transportation over advertising via papers;
- "Present and Offer" allows customer to browse and select purchases online instead of through paper catalogue or in store, which helps to reduce cost for storage and transportation of goods, as well as energy consumption for maintaining warehouses;
- "Select and Order" tends to improve environment by using secure email for negotiation instead of papers;
- "Trust and Pay" focuses on the reliability in payment process;
- "Deliver and Trace" allows users to monitor current state and location of the purchased item, as well as considers the delivery impact itself;
- "Customer Relationship Management" focuses on social sustainability, which tends to improve customer satisfaction.

The idea that e-business is conducive to sustainability is also advocated in [8], where the authors note that the development of such technologies as cloud computing, mobile communications and social media helps to advance sustainability of a business and contributes to societal development.

C. Measuring Impact of Sustainability Requirements

In order to measure the impact of sustainability-related requirements of the engineered system, one needs to utilize some evaluation metrics. Here two distinct perspectives can be adopted: one focusing on the sustainability of software itself (i.e., software quality), the other on the impact that this software has on the broader environment (ecologic, economic, and social) within which it is situated. The prime interest of this study is in broader impact of the software system.The present study aims to utilise the pre-existing metrics to the highest possible degree, only providing additional metrics where, to the best of our knowledge, no previous metric exists.

When evaluating the impact of the software on the business and its wider environment, the related work suggests focusing on the system effects in terms of energy consumption or CO_2 emission. For instance, [7] suggests a conversion factor between gasoline, CO_2 and energy consumption; [9] points out the linear relationship between CPU usage and power consumption. And [5] provides a detailed reference on carbon footprint measurement for a multitude of products and activities, ranging from purchase of a banana, to sending SMS, etc. We utilise the estimates from [7] and [5] as reference measures in evaluation of the impact of the environmental and economic requirements in terms of the amount of energy consumption. However, impact of the requirements related to social sustainability is much more difficult to evaluate. Some such requirements need a long verification time, as their effects will surface only on future generations (Korte et al. [8]), others require access to system users. Since the system in our case study is not in real use, we have no means of collecting data for evolution of its social sustainability effects. Thus, although a number of social sustainability requirements are identified and discussed in this study, we do not conduct evaluation of system's social sustainability impact.

Thus, the contributions of this paper can be summarised as:

1) presentation of an end-to-end re-engineering study where sustainability requirements are identified, implemented for a pre-existing system, and evaluated;
2) presentation of a set of metrics suitable for quantification of environmental sustainability in requirements;
3) demonstration of the measurement process of environmental sustainability requirements of a software system;
4) demonstration of the application of the consolidated research on software engineering for sustainability.

Next, Section II of this paper presents the Pet Shop case study. It is followed by discussion of the methods and metrics used in this study in Section III. The sustainability requirements and their mutual dependencies and contributions to environmental impact are discussed in Section IV. The details on the actual and expected impact due to system re-engineering are presented in Section V. We then discuss several points relevant to this study in Section VI.

II. Overview of the PetShop application

Java Pet Store 2.0 is a Reference Application [10] provided by Sun Microsystem as a demonstrator for J2EE. It is a web application that allows customers to search, buy, and sell animals online. Several factors were considered in choosing this application for our case study. Firstly, as noted above, e-business is becoming an increasingly widespread medium of business conduct with an ever larger portion of transactions undertaken over the internet, which makes this domain worthy of attention. Secondly, as a reference application provided by Sun, Pet Store should minimise (and ideally avoid) any potential findings caused by poor system design or inappropriate use of programming language. Finally, the application is freely available to general public, in case others wish to replicate or extend out study.

The Pet Store application provides basic e-shop functionality without concerning the sustainability requirements. The supported functionality is shown with white and yellow coloured use cases in Figure 1 (use cases in red are to be discussed later). The Pet Store documentation states that:

- Seller can upload information on a pet (including its name, description, price, image and seller's contact information) to a specific category. The seller's contact requires last name, first name and address. An image thumbnail will be created corresponding to the uploaded image;
- Seller can tag an item;
- Web user can locate items by choosing category or searching them by tags, location or keywords;
- When viewing an item, web user can also rate it in range from 1-5.
- Web user can set an item as inappropriate;

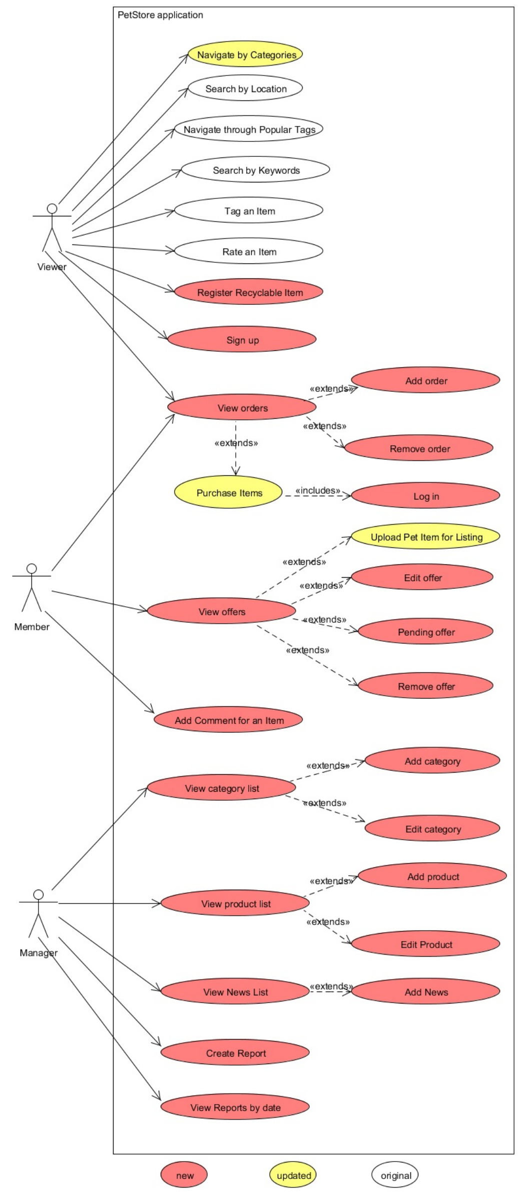

Fig. 1. Use cases with sustainability integrated

- Buyer can buy one item at a time;
- The application can get and show RSS news of Java Blueprint on the top of the page;
- The application assumes that there is an existing list of categories with their appropriate images such as "Cats", "Dogs", etc. Besides, each category has a list of products with their appropriate image such as "Medium Dogs", "Small Dogs", etc. An item is categorised according to products.

As shown in Figure 1, three types of actors are distinguished:

- A viewer, who is an ordinary web user and can access the PetStore site;
- A member is a viewer who has signed up and logged in into the system;
- A manager administrates the PetStore application. Manager has some unique behaviour, e.g., creating and viewing reports (as defined in Figure 1) but also has the same access and use rights as a member.

The colour of use cases in Figure 1 is used to distinguish the originally defined and modified use cases:

- use cases which have been defined in the original system and remained unchanged throughout the below discussed system re-design are not coloured (i.e., are white);
- the yellow use cases were defined in the original use case model, but their behaviours had to be changed due to the newly defined activities/requirements for improved sustainability;
- the red cases are the newly defined use cases, that are to be discussed in following sub-sections.

III. Methodology and Metrics

In order to undertake the intended system re-design study, we must define a process for relevant requirements identification as well as a mechanism for comparing the updated and original versions of the system. These are outlined in the following sub-sections.

A. Requirements identification and modelling

As discussed previously, a number of researchers ([3], [4]) describe successful use of goal-oriented technique for identification of sustainability requirements. Following their lead, we too utilise this approach. This is particularly suitable for our case study as it involves re-designing an existing application, where the other goals of the application are already defined. Unlike the cases of customer-commissioned systems, for the purposes of this case study, we have no direct access to those commissioning the PetStore system. Thus, instead we rely on the goal decomposition structures that have emerged from the relevant previous studies, i.e., the key drivers for e-business success discussed in [7] that are directly relevant to economic, social, and environmental sustainability, as well as the reduce-reuse-recycle goal set aimed at environmental sustainability derived by [3].

Using these generic goal structures as starting point, we instantiated them within the context of the PetStore application and operationalised them into tasks. These tasks (also shown as red and yellow use cases in Figure 1) have then been integrated into the updated system design and implementation.

To represent the goal decomposition and contribution models, we use the i* language to capture the context as well as relationships between sustainability and other system goals [11], as i* is particularly effective in modelling the problem domain in the early stages[1]. Moreover with i* we can model both hard (i.e., functional) and soft (i.e., non-functional) goal structures, as well as represent the links between goals and their respective actors - all of which is useful for the present study.

B. Metrics

As pointed out above, our prime interest is in studying how consideration of sustainability requirements in a software system impacts the sustainability, i.e., the broader (ecologic, economic, social) environment within which the software system is located. Thus, we are interested in measuring environmental impact (i.e., impact achieved due to use of software) in terms of emissions.

1) Measuring Environmental Impact: The **environmental impact** of the re-design of the system can be measured by comparing its CO_2 emissions to emissions from the original system.

In particular, it can be estimated using the two following facts:

1) Fact 1: The amount of CO_2 emitted (CO_2e) for a unit of electricity from UK grid[2] is about 600g [5]. This includes both direct emissions (i.e., from power generation per unit consumed), and indirect emissions (i.e., including the carbon cost of extracting fuel from ground, maintaining the buildings for the power stations, producing wind turbines, etc.).

2) Fact 2: Empirical measurement of power use with various types of servers has demonstrated a stable linear

[1]We used UML for later stages or detailed design.

[2]It is worth noting that this emissions number is very country-specific, ranging from 60g. for Iceland to 1060g for Australia. The exact value depends on the fuel mix used by the given country's grid with Iceland using only natural geo-thermal sources where other countries use a mix of sources, such as fossil fuels (oil, gas, and coal), nuclear power, renewable, etc.

relation (accurate to within 5%) across all CPU utilization rates [12]. This relation is shown in the equation below, stating that the energy consumed by a CPU for a particular task (P_x) is made up by the sum of the amount of energy the CPU consumes in idle state (P_{min}) and the CPU utilisation rate (x as percentage) times the difference between its idle and fully occupied state (P_{max}-P_{min}) divided by 100:

$P_x = P_{min} + x*(P_{max}-P_{min})/100$

Where :

P_x: Power consumption for x%CPU (watt)

P_{min} : energy consumption in idle state (watt)

P_{max}: energy consumption for 100% CPU usage (watt)

For example (as per [12], if a server has a maximum power draw of 300 watts (W) and an idle power draw of 200W, then at 5% utilization the power draw would approximate to: Power Utilization at 5% = (300 - 200) * 5/100 + 200 =100 * 0.05 + 200 =205W

Thus, the [$x*(P_{max}-P_{min})/100$] component in the above equation marks out the additional energy used for the completion of a given task (on top of the idle state consumption). Since, as per Fact 1, each unit of electricity from the UK grid produces 600g of CO_2e , the CO_2e for a task (CO_x) can be calculated in *kilograms per Watt* as:

$CO_x = 0.6 * (P_x - P_{min}) = 0.6 * x*(P_{max}-P_{min})/100 = 0.006*(P_{max}-P_{min}) *x$

To account for the duration of a given task, this formula needs to be further adjusted for the time. Thus, the emissions (in kilograms per Watt) for a task with a given duration in seconds (T_x) will be calculated as:

$T_x = CO_x*3600/ C_{average}$ seconds

Where:

$C_{average}$ is the average CO_2 emission (CO_2e) in kilograms per hour

CO_x is CO_2e (in kilograms per Watt) for doing the task.

2) Implementation of Metrics Data Collection Framework: In order to utilise the above discussed metrics, the study needs to collect the relevant data. Data for environmental sustainability requirements is to be evaluated based on CO_2 emission and task processing time. CO_2 emission for a task is measured as $C_x = 0.006*(P_{max}-P_{min})*x$. And the appropriate processing time for a task is approximately $C_{exp}*3600/ C_{average}$ seconds in which C_{exp} is the expected CO_2 emission and $C_{average}$ is defined according to testing system.

Therefore, to collect data for measurement of per-task emissions, CPU usage on server side and processing time on client side should be monitor throughout processing of a given task. The data collection set up is illustrated in Figure 2. For the present case study, a Java application was implemented using Sigar library [13] to get the CPU percentage of server use during a given task. The application records the data into a log file on server side.

On the client side, we use the Firebug [14] tool to log the data on the network transport, including number of requests, total size, total of loading times, processing time per request, in a HAR file. This is used to obtain the processing time of a given task (T_x).

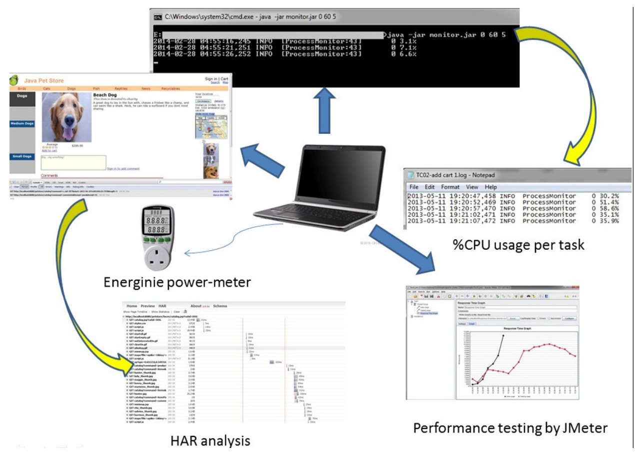

Fig. 2. Data collection setup for Pet Shop

We use black box testing to verify the implementation of our re-design for sustainability by defining a test scenario for a task to be evaluated, recording the C_x and T_x for each task in both original and updated versions, and comparing these. Where a task to be evaluated is not supported by the original version of the system, the data on updated system is compared against an impact value obtained from an alternative buy comparable way of conducting the required task without the system support (further discussed for each relevant task in section IV).

IV. Sustainability Goal Analysis for Pet Shop application

Having outlined the goal-based analysis process and metrics to be used for case study evaluation, we now present the decomposition and operationalisation analysis of sustainability goals. Since the combination of all goal models on sustainability generates quite a large and complex picture of requirements, below we introduce the three sustainability views one at a time. For each of the discussed views the task operationalisations are also discussed in terms of their expected environmental impact.

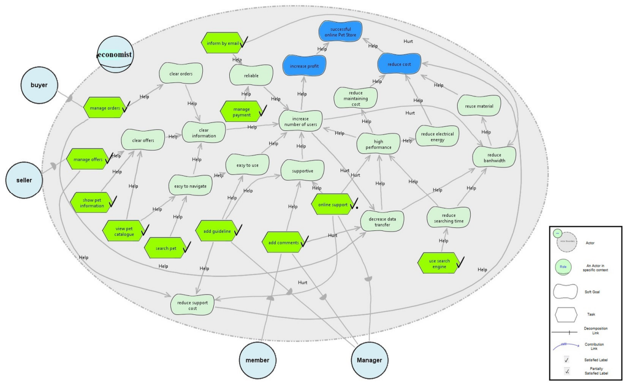

Fig. 3. Economic sustainability.

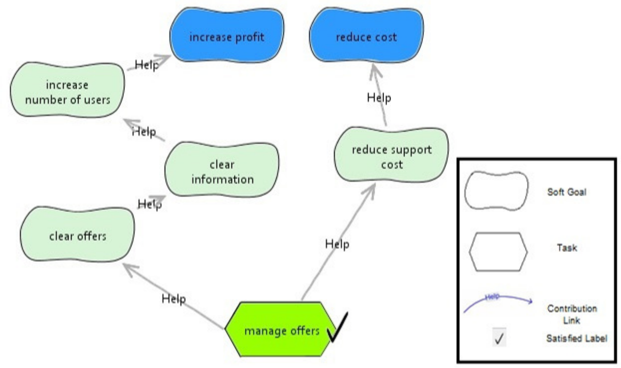

Fig. 4. Task: Manage offers

A. *Economic sustainability*

In economic sustainability view the emphasis is on two main goals: increasing profit and reducing cost. Figure 3 demonstrates the decomposition and operationalisation of these two goals.

For e-business such as PetStore application, one of the most important factors supporting its' economic sustainability is the number of users accessing the website (shown as "Increase Number of Users" goal in Figure 4). In other words, the more users use the website, the more potential benefits it will achieve. In accordance with [7], this goal can be achieved via clear information presented on the site, good usability, support and performance of the site, all of which are depicted via the goal contributions in Figure 4. Some other core goals related to cost reduction are, for instance, decreasing maintenance costs and reusing materials. Figure 4 shows several tasks (in green) aimed at operationalisation of the discussed goals (such as manage offers, manage orders, etc.), which are discussed below:

Manage offers task allows sellers to view/add/remove/edit their uploaded pet items. The current system allows only uploading information, without support for further review and change to the uploaded content. This is an obvious shortcoming in terms of system usability for clear information presentation to the seller. With the additional functionality to mange the sale items, a seller can track the status of his/her items, change and upkeep sale details. This also helps to reduce website support cost: instead of the seller sending change requests to the site administrator, he/she can make the necessary changes by him/herself. The contribution chain of this task is depicted in Figure 4.

Sustainability impact measurement: In the original version of the system, where this option was absent, whenever a seller needed to make a change (e.g., change name or price of the offered item), he/she had to send an email to the system requests, asking the administrator to make the change. After receiving the email, the administrator would browse for the specified item and apply the requested change. Therefore, to improve system's impact, the CO_2e for making a change (C_x) with the updated version of the system should be less than the sum of CO_2e caused by a i) sending a change request email and ii) carrying out the change by the admin.

According to Berners-Lee [5]:

- a single (non-spam) email costs about 4g of CO_2e (including sending, filtering and reading);
- CO_2e of a web search on the local network and host server is about 0.05g
- CO_2e of systems' hardware required for a web search reaches up to 0.35g per search.

Let's assume (for simplicity) that the only difference in handling the change in the original and updated systems are in sending email for administrator to make the change vs. doing a web search to implement the change in person by the seller (after which both sides will presumably use the same web interface to carry out the change). Then, to improve the system's environmental impact, C_x for seller should be less than 3.6g (i.e., 4g. i.e. (0.05g+0.35g)). Moreover, as an average user takes about 30 seconds to search, wait and scan the result from the web search [5], T_x must be less than 30 seconds.

Manage orders task allows buyer to view/add/remove/buy pet items. With the original

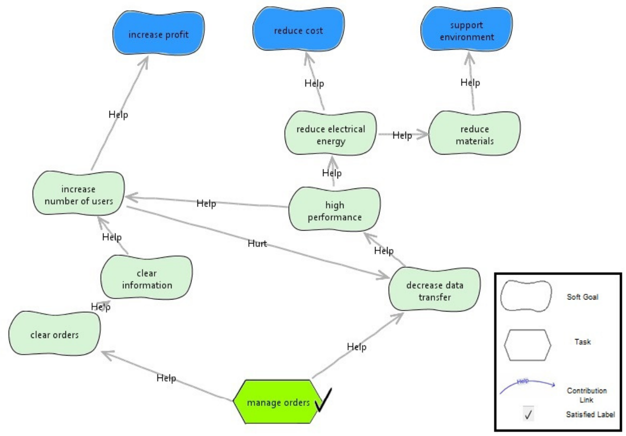

Fig. 5. Task: Manage orders

system a buyer can buy only one item per transaction. The extended system will allow the buyer to reserve several items while shopping, and then process the payment for all reserved items at once. For multi-item purchases, this update will both reduce the need of repeated data exchange and payment processing between the customers and server, and make shopping process easier. As shown in Figure 5 below, this change should result in reduced energy consumption (due to reduction in payment request processing). On the other side, we observe a negative contribution between the "increase number of users" and the "decrease data transfer" goals. However, this is unavoidable as attracting more users will lead to the need to process more data. This tradeoff must be resolved through discussion with the owners of the system.

This task can be operationalized via addition of **"Add to cart"** and **"Card"** links to the system.

Add to Card allows buyers to add item to a cart. The item is marked as "Booked".

Cart link allows buyer to view/remove/buy items.

Sustainability impact measurement: the original version of the system supports only buying one item at one time. To improve sustainability, the updated system must, at least, deliver the same performance, as use of the cart will, at minimum, improve the shopping experience (i.e., social sustainability). This means that the CO_2e and time processing for buying a single item in updated system must not be greater than that of the original version. Moreover, the summative power and time consumption for buying n items at once has to be less than that of buying n items one-by-one. This means that C_x should be less than one caused by the original version and T_x should be less than spent while using the original version.

Add guideline task assists customers in understanding how to use the web application. The guidelines are provided by the site Manager. This task helps to ease the site use via clear information presentation, which is conducive of user number increase. The task contributions are shown in Figure 6.

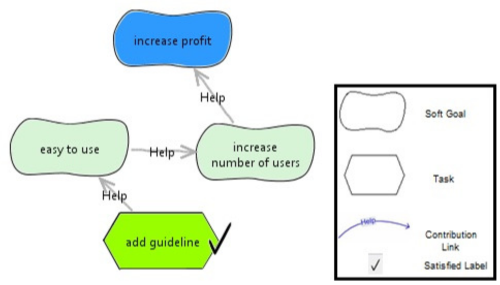

Fig. 6. Task: Add guidelines

Sustainability impact measurement: this task will increase the environmental impact of the system by the value of CO_2e for web search and review of the help pages, but will decrease the emissions by the value of incorrect system use (e.g., mistakes in sale uploads and related correction requests, etc.). The net impact value of this task is very dependent on the help content and system use patterns.

Inform by email task informs the corresponding buyers and sellers on the payment processing success or failure. This is a necessary function to build the users' trust in the business' service. The impact of the task is two-fold: it increases the perceived reliability of the web application, but also increases CO_2e emissions due to extra emails. The contribution chain for this task is shown in Figure 7 below.

Sustainability impact measurement: this task will increase the environmental impact of the system by the value of CO_2e of an email for each purchase. On the other hand, the improved trust into the system by users is not easy to measure, especially in the present case study where the system is not in actual use.

Online support task provides quick on-line help to the customers, improving their shopping experience. However, this will negatively affect the application performance due to use of real-time technology, which also increases energy consumption. Besides, the real-time support leads to additional costs required for continuous

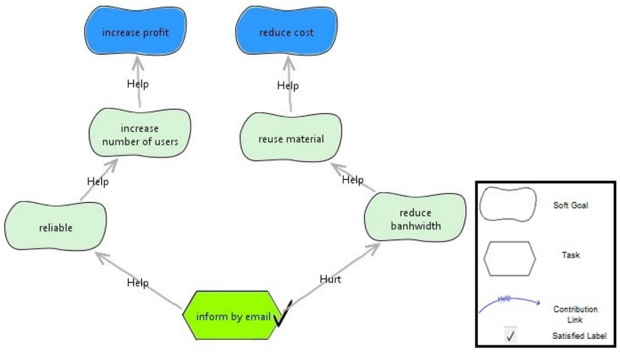

Fig. 7. Task: Inform by emails

presence of an employee to provide the requested help. The contributions for this goal are shown in Figure 8 below.

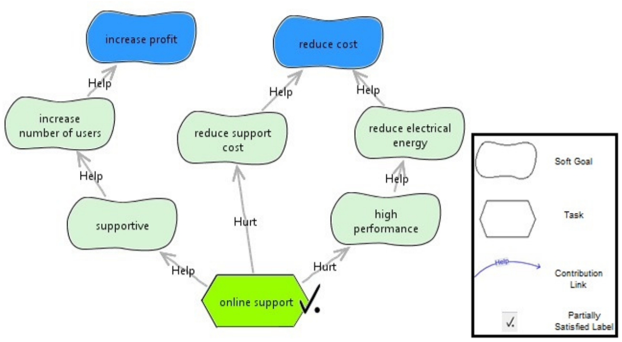

Fig. 8. Task: Online support

Add comments task allows users to add comments to each pet's items or to news items. This is a mechanism for user-initiated discussions as well as mutual support/help between Pet Shop customers with regards to specific items. This will improve shopping experience, make the website more interactive, as well as more socially inclusive (and so, more socially sustainable). On the other hand, the energy consumption for data exchange and storage will increase, as shown in Figure 9 below.

This task can be operationalised via addition of the **"Comments"** option that allows users to add/edit/remove their comments for each item.

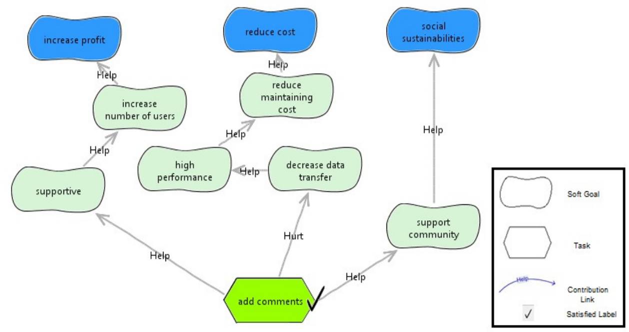

Fig. 9. Task: Add comments

Sustainability impact measurement: Although this function increases CO_2e due to posting on the web site, it is also very useful in attracting more users and supporting community. On the other hand, it can indirectly support users' query resolution. In absence of comments, an email has to be sent for each query (e.g., to administrators, or sellers) with the cost of 4g CO_2e per email, out of which 0.4g CO_2e is caused by the internet infrastructure [5]. Thus, CO_2e for posting a comment (C_x) must be less than 3.6g (i.e., 4g - 0.4g as comments too use web infrastructure). Moreover, the comments function displays the posted comments on the web site right after a member submits it, which is similar to the process of sending, receiving and reading an email. Thus, T_x should be less than 30 seconds (which is that spent for sending, receiving, and reading an email as per [5]).

B. Environmental sustainability

For environmental sustainability we also adopt the overall "reduce-reuse-recycle" perspective advocated by Cabot et al.[3], as, reuse and recycling of already produced goods economises on the need for new production, while reduction of resource consumption supports longer-term availability of restricted resources as well as reduces the overall impact (emissions, extraction, soil degradation, etc.) of resource overuse.

In context of PetStore application, reuse and recycling of other products made for pets and pet owners (e.g., carrier cases, accessories, etc.) were considered. Figure 10 presents the goal contributions identified for the environmental sustainability of the Pet Store, with a number of tasks operationalisations discussed below.

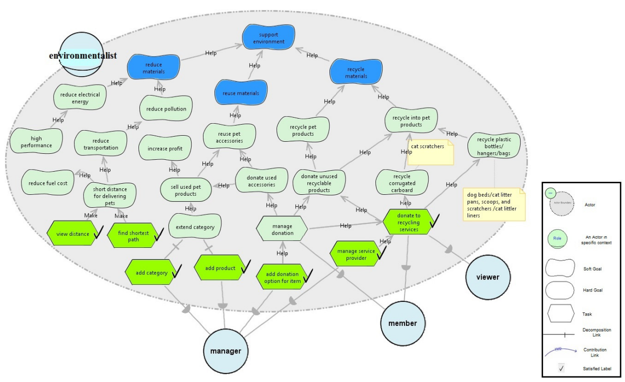

Fig. 10. Environmental sustainability

View distance and ***Find shortest path*** tasks aim at reducing pollution caused due to transportation. *View distance* shows the direct-line distance between user's given location and the location of the item-to-be-purchased. *Find shortest path* provides travel directions for the route between these points. These functions are useful for both buyers and the Pet Shop business. The buyers may be choosing to purchase in environmentally-friendly way (with transportation emissions in mind), or planning on collecting items in person. The Pet Store will also use this for choosing their delivery and/or collection routes. The contribution structure of this task to the goals is shown in Figure 11.

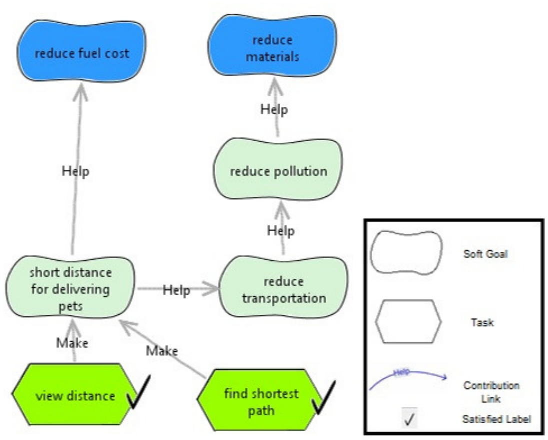

Fig. 11. Task: View distance and Find shortest path

This task can be operationalised via addition of the **"View Distance"** option.

Sustainability impact measurement: As a single search causes about 0.7 CO_2e which includes 0.1g for searching, waiting and scanning results, C_x for this task should be less than three searches, i.e., 3*0.6g = 1.8 g.[5]. This is because a similar search can be done by e.g., using AA route planner, but only after an additional search for finding such a route planning site. So, adding this functionality should be better than finding the same service via an extra web search. Which means: one search for finding the link to a given page, then loading it, typing in the start and end points for the travel and, finally, loading the suggested route page. In other words, T_x has to be less than 3*0.6*3600/ $C_{average}$ seconds .

Add category and ***Add product*** tasks allow for the PetStore to sell not only pets but also related pet products. In practice, when a user sells a pet, he/she often wants to sell the pet's accessories as well. This functionality then helps to reuse/re-sell pet products. The original application supports selling pets only. Thus, this functionality helps in both reusing items (reducing emissions for new item production and improving environmental impact) and in extending the business to other related markets, which attracts more users and improves profits for the business, which is shown in Figure 12 below.

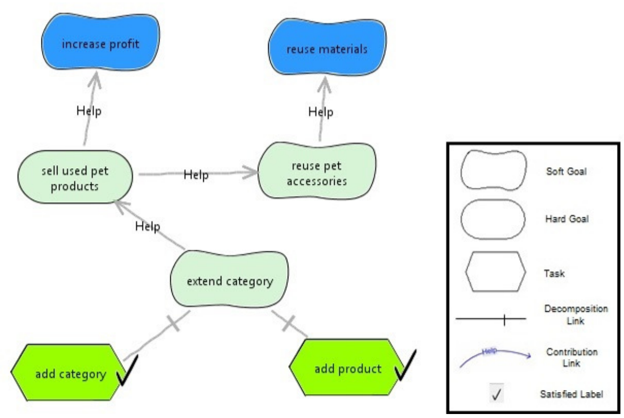

Fig. 12. Task: Add category and Add product

These tasks are operationalised by providing the **"Add Category"** and **"Add Product"** options for new categories and products respectively.

Manage donations task allows the seller to donate the pets and/or items to a particular charity. If this option is used, the Pet Store will receive payment from the buyer purchasing the donated item and transfer it to the charity selected by the seller. This will not only help with reusing pet items but also support the community at large.

This task is operationalised via **“Donate”** option which allows users to donate items to charity. Such items are marked as “donated”. The contribution structure of this task is shown in Figure 13.

Sustainability impact measurement: Because this is an additional choice option added on top of the already present functionality of uploading an item for sale, it is necessary to maintain or improve the performance of the original version. Besides, this function supports social sustainability, by promoting positive, helping attitude in the participating Pet Store user community. The social impact could be measured via responses from users over time, but that is not a part of the current study. For the purposes of this study we expect that C_x and T_x should be no higher than those of the original version.

Another utility of used items is in ***Recycling***. Recyclable products can be donated for material reuse. Thus, recycling items can be listed by customers, collected and processed by recycling service providers such as factories. Pet Store will donate the profits from the recycling contributing to social well being, as well as improve environmental impact via reuse of materials. In general, although the energy consumption of the Pet Store system will increase due to more tasks to be processed, this task will reduce the overall pollution if it is well integrated with the “find shortest path” task discussed above. The contribution structure of this task is shown in Figure 13.

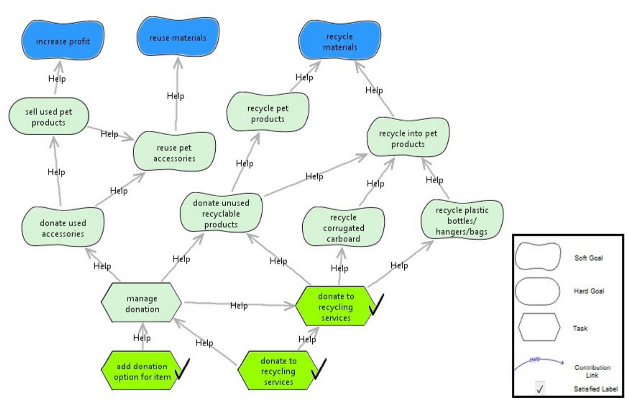

Fig. 13. Task: Manage donations

This task is operationalized via **“Register recyclable items”** option that allow users to upload recyclable pet items into relevant category (e.g., Recycle into pet products, corrugated cardboard, plastic bottles, hangers, bags) for donation.

Sustainability impact measurement: This option carries out a transaction for donating used items of a given weight (kg). According to [5], CO_2e caused by 1 kg of trash put into landfill for decomposition is 700g. While a web base submission causes only 0.6 g. of emissions. However, submission of recycling request via this task, is only the first step for further processing of submitted goods, and will require such further tasks as collection/transportation and actual recycling processes. These will, in turn, produce extra CO_2e. The actual amount produced will depend on the distance between source and destination points for collection and delivery, as well as the recycling technology. However, considering that:

1) if submitted to waste, these items would still be collected and delivered (though to landfill instead of recycling plant), and
2) the cost in CO_2e of virgin (instead of recycled) production is generally much higher for most items (including aluminium from cans and plastic from bottles as per [5]),

We are confident that the impact of recycling is normally less than that of landfill. Thus, in this case, the CO_2e caused by this processing must be less than a one-off single request such as a web search without waiting and scanning result. Hence, C_x should be less than 0.6g and $T_x < 0.6*3600/C_{average}$ (s).

C. Social sustainability

In terms of social sustainability, we review the requirements that aim to achieve customer satisfaction and knowledge exchange in society. Both of these tasks help in contributing to long-term social well-building. The goal model for social impact is shown in Figure 14, with a set of relevant tasks analysed below.

Add news task allows site manager to add/edit/remove news. News are articles related to pets, such as pet policies, pet care or training programs.

This task is operationalised by provision of the **“Add news”** option to the site.

Sustainability impact measurement: This task must cost less than a printed latter or an advertising email which carry out the same function as posting news articles. Since cost of a non-spam email is at 3.6g - to 50 g., excluding emission because of internet infrastructure is lower than that of printed latter (140 g), CO_2e for processing this task should be less than sending an email. As one may note, while in previous discussions we have used the lower limit of the email-motivated emissions,

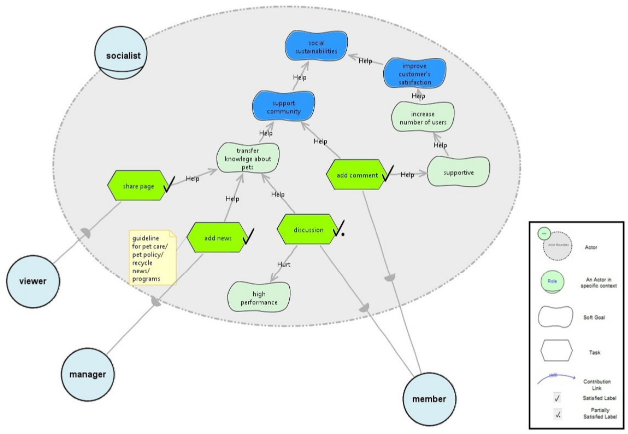

Fig. 14. Social sustainability

in here we would consider a somewhat higher value, as news articles would normally be longer than a standard email. It is known [5] that the CO_2 emissions of an email is dependent on the email's length and attachments. For example, processing a long email with large attachment will emit up to 50g CO_2. Thus, putting a rather restrictive target, we would aim to keep C_x to about 10g CO_2e and the processing time T_xe should be less than 10*3600/ $C_{coverage}$.

Share page task allows users to share pages on social networks such as Facebook, and Twitter. It not only helps the business reach potential users but also helps to share knowledge in interested communities, e.g., how to train a pet, or care for it.

Sustainability impact measurement: The effect of this task comprises of submitting a single 'share page' request to a social network site. Thus, CO_2e caused by its processing must be less than of a single web search (i.e., 0.7g.) which would otherwise be needed (by, e.g., a Facebook user) for searching for and finding the information on a page of interest.

Support discussion: This task allows users to create and discuss specific topics. It gives a place for members to exchange knowledge. However, this task will require extra effort for managing and exchange of data, which can impact on performance of more prominent functions such as selling items.

Reporting allows managers to view sustainability-related data, such as profits, donation rate, rates of selling recycleable and reuseable items, CO_2e emissions, number of members at a given time, etc. Strictly speaking, this task did not originate in the goal analysis patters we chose to instantiate. Instead, it surfaced from re-design process. This task eases the managers' work through coherent information provision.

Sustainability impact measurement: To record the relevant information, CO_2e must be less than those for using computer in 10 seconds, which is about how long it will take to submit a report request, get the results back, and quickly scan them. This means that C_x <$C_{average}$/360g and T_xe <10 (s).

V. Result Evaluation

A. Environmental Sustainability Evaluation

To evaluate sustainability requirements, task simulation was set up on a laptop with the following configuration: Gateway Intel(R) Core(TM) i5 CPU M430 @ 2.27GHz, 4GB RAM. A server was deployed on Glassfish 3.1.2 run with Java EE 6.0 and built-in Derby database. On client side, the chosen browser was Firefox 20 because it supported the Firebug plugin which is used to collect data to measure the task processing times.

As discussed in section 3:

1) For calculating CO_2 emission corresponding to CPU usage, it is necessary to determine maximum and minimum power usage of the computer used. We used a power meter Energenie [15] to monitor the maximum and minimum value of the laptop's power consumption within one hour. We observed the maximum power usage P_{max} of 60 Watts and the minimum power usage P_{min} of 20 Watts.
2) The CPU usage for a task is the difference between CPU usage monitored during the task and the average CPU usage in idle mode where the laptop is connected to internet but is not doing any specific task. Thus, CPU usage in idle mode is observed for 10 minutes before carrying out a task (for both original and new versions), in order to obtain the average value for the idle mode before each data collection task.
3) If both original and new versions of the system support a given scenario, C_{exp} is taken to be the amount of CO_2 emissions caused by the original version. However, if the original version does not support a given scenario, the C_{exp} is the value calculated for an alternative but comparable way of handling the identified task/requirements. The CO_2 emission of the updated version should be less than this calculated value to improve sustainability, as defined in section 4.

Another parameter used in this evaluation is time of processing (T_{exp}). In the scenarios supported in both versions of the system, time of processing of new version is compared with that of the original version. Where a task is newly defined, and so absent from the original system, T_{exp} is defined according to $C_{average}$ and C_{exp}. $C_{average}$ is amount of CO_2 emitted by an efficient laptop in an hour, which is set to 25g as per [5].

Thus, in order to demonstrate improvement in the system sustainability, the CO_2 emissions (C_x) for the tasks defined in the updated version should be less than C_{exp} and the processing time (T_{exp}) has to be less than C_{exp}*3600/$C_{average}$ seconds.

B. Sustainability evaluation Results

Table I shows the values collected from running a selection of the scenarios from the above discussed tasks. Scenarios used for evaluation of the updated system through simulation are those for which we could collect comparison data either from the original version, or by building a comparable estimate form alternative tasks. Since the present system is not in actual use, we had no access to users to conduct any social sustainability comparison, thus all such scenarios are left out. The summative results show in the Summary row of Table I demonstrate that the updated version of the system reduces overall emissions by about 25% and the overall required processing time by a factor of 28. These results are obtained with substantial new functionality added to the system.

The above results demonstrate that the actual values are not always within the bounds of the expected ones.

The TC01 scenario of the *Edit an offer* task has CO_2e of 3.13g which, as expected, is less than that caused by requesting changes via email. Thus, this function helps users to save time and resources. Also, the time processing for this scenario is around 30% of expected time. Although the test is run over a local network this time efficiency suggests that even when running over internet, the processing time should take less than 30s.

In the TC02 scenario of *Add item to cart* task the buyers have not yet requested to buy an item, but have placed it into a cart, which gives them a change to review their decisions and expenses, without forcing the purchase. This is a positive contribution to the shopping experience. The amount of CO_2e of this task is about 30% of the respective processing in the original version which is accessing Paypal service without processing any payment (estimated to 4.26 g). The time used for processing, however, is nearly double that of expected, which could be explained by the need for the application to access the database, mark the chosen items as booked, and return.

In scenario TC03 to buy 1 item through *Cart* the server processes the purchase of a booked item from the member's cart. In the original version, this action directly connects to Paypal service (but does not carry out processing due to security concerns). In the new version it instead connects to the server to simulate payment. In terms of CO_2e both versions are similar, with the updated system causing approximately 74% of the original one's emissions, and so showing some improvement of application. However, it has a longer processing time (5 seconds above the original system's). This is caused by the need to enter shipping address and create a payment (coming to 3.5 seconds), as well as the additional information provided in the new layout display, which takes additional time when a user browses the catalogue. Yet, it must be noted also that the original version doesn't allow for simulation of an actual payment processing, so the time taken for this task in that version is likely to be under-estimated.

The TC04 scenario extends the previous one through *Cart* scenario with purchase of more than one item. We observe the amount of CO_2e emissions is less than that of the original version. Moreover, there is only 9% increase (from 74% to 83%) when buying 5 items instead of 1. However, processing time is more than 4 times of the expected. Looking at this time in more detail, we note that it takes about 2 seconds between booking each item, while in the original version the in-between time is shorter because a new window is opened for processing purchasing, which was not taken into account in the previous measurement. Moreover, as in the case of scenario TC03, the original version does not support actual payment processing simulation, while the updated one does.

In TC05 scenario for Add a comment task the length of the comment is limited to 255 characters, which is comparable to an average (non-spam) email. The emissions and time used results include monitoring the whole process (i.e., signing in, posting and reloading comment list). Therefore, 50% reduction in emissions and large reduction in processing time (over 8 times) are good results.

The TC06 scenario of *View distance* task emits nearly exactly as much CO_2e as 3 single searches set as expected value, with only 0.1 g. more. One of the main reasons for this extra bit of emissions is that the system connects to Google Maps service and downloads images which usually take longer to load than text. Moreover, this task also helps to improve social sustainability. For example, user now can visualise item's location and

Task Scenario and Steps	**C_{exp}**	**T_{exp}(s)**	**C_x(g)**	**T_x(g)**
TC01: Steps for *Edit an offer*: • Logged in successfully with correct username and password • Click Offers > Edit for an item	3.6	30	3.13	5.94
TC02: Steps for *Add item to cart*: • Browse for a pet item • Click "Add to cart" • Repeat for original version: browse for a pet item then choose "Buy now"	4.26(*)	4.5(*)	2.91	7.41
TC03: Steps for *Cart for buy 1 item*: • Logged in successfully with correct username and password • Browse a pet item • Click "Add to cart" > Click "Cart" link • Enter shipping contact • Choose "Buy now" • Repeat for original version: browse for a pet item then choose "Buy now"	2.88(*)	1.42(*)	2.12	6.73
TC04: Steps for *Cart for 5 items*: • Logged in successfully with correct username and password • Browse for a pet item • Click "Add to cart" > Click "Add to cart" • Repeat above steps 5 times • Enter shipping contact • Choose "Buy now" • Repeat for original version: browse for a pet item then choose "Buy now", do this for 5 items	2.37(*)	4.6(*)	1.99	19.51
TC05: Steps for *Add a comment*: • Browse a pet item • Submit comment "This is TC05"	3.6	30	1.94	3.57
TC06: Steps for *View distance* • Browse a pet item • Enter zip code "94105" to "Your location" • Click "Get distance"	1.8	259.2	1.81	3.4
TC07: Steps for *Add news* • Logged in successfully with correct username and password • Click "Admin" > "News" tab > "Add" • Enter required information for news • Submit • Browse the news	10	1440	4.36	10.89
TC08: Steps for *Support donations* • Logged in successfully with correct username and password • Click "Offer" > "Add new" • Enter required information for a pet item • Select "Donated" option • Submit • Repeat for original version (without "Donated" option)	2.96(*)	13.96(*)	2.62	6.95
TC09: Steps for *Register recyclables* • Click "Recyclable" • Specify "Weight" is 1 kg and other required information • Submit	0.6	86.4	2.99	4.80
TC10: Steps for *Create report* • Logged in successfully with correct username and password • Click "Admin" > "Report" tab > "Create Report"	0.07	10	1.27	1.13
TC11: Steps for *Load reports* • Logged in successfully with correct username and password • Click "Admin" > "Report" tab > "Create Report" • Select "From" and "To"	0.07	10	0.73	1.61
SUMMARY	32.21	1890.08	25.87	64.53

TABLE I: Evaluation results for sustainability requirements [3]

[3](*) Value got from original version

estimate distance and cost of travel before making a decision on purchasing or transportation. It also improves awareness on CO_2e emissions to environment caused by transportation, as well as reduces processing time by factor of about 76.

The TC07 scenario for Add news evaluates to very effective results with the emissions coming to under 50% of the expected value and over 100-fold improvement in processing time.

The TC08 scenario for *Support donations* task extends the original version (with marking selected items as donations) with additional slight reduction in resultant emissions. The processing time for uploading donation items is also twice as quick. The time and emissions are reduced since the user has to sign in with the updated version but must type in his/her contact information for every new submission with the original system (as it does not support user accounts).

While the TC09 scenario for *Register recyclable*s task takes much less time than expected, it causes 2.99 g CO_2e emissions - about 5 times greater than a single query (which we use as the expected emissions target). Yet, as discussed in previous session, the overall environmental benefit from this task is likely to be much greater due to reduction in virgin resource extraction demand. This, however, is not evaluated in the present study.

The TC10 *Create report* and TC11 *Load reports* scenarios for the *Reporting* task both have much higher emissions than the 10 seconds use of an efficient laptop. One reason for this is that to create a report the administrator has to access the admin page on the web and select the categories, products and articles which are listed on it. This takes effort from both server and client to process. However, the actual emissions values are rather small and the processing time is also short. Since, these functions support monitoring sustainability benefits, which helps the manager to evaluate and plan further actions, all of which is useful to a system.

VI. Discussion

Validity of Results: While all code and experimental data presented in this paper are freely available for cross-validation, there are two issues that are worth pointing out with regards to the validity:

- The estimates for CO_2e emissions are largely taken from [5] where they are calculated using the currently accepted carbon calculation methodologies. However, it is worth noting that alternative carbon estimation techniques are also well used and different techniques often lead to different estimates.
- When choosing an expected emissions level (in section 4), we have consistently looked for lower-bound estimates which the updated system would have to fit into. Thus, it is very likely that the carbon reduction from the presented redesign is in fact larger than what we have reported it to be.

Tensions within Sustainability: The findings from this case study demonstrate that there is an ever present tension between the three "pillars" of sustainability, i.e., its environmental, economic, and social sides. This is because economic sustainability for a business supported via software aims to increase consumption, while environmental sustainability aims to reduce it. The social sustainability, in turn, has a conflict with itself - aiming at both increased, faster, and more convenient consumption available to all and vibrant, un-spoiled natural environment at the same time. Yet, as it stands today, increased consumption inevitably lead to environmental depletion. This conflict has been observed within several discussed requirements (e.g., should the system provide continuous on-line support at increased economic expense? should the system increase number of users at the cost of increased emissions?, etc.). While conflicts within system goals/requirements are quite commonplace in all development projects, there is an additional diminution to sustainability conflicts: since social and environmental dimensions are "invisible" , until recently, the monetary returns have massively biased trade-off resolutions in favour of income maximisation. It is now necessary to educate software professionals on the value of the invisible environmental and social dimensions. This does not mean that businesses must always lose income. On the contrary, many environmentally or socially motivated requirements will turn a profit. For instance, in Pet Shop, registration for collecting recycled products, not only helps to reduce a large amount of landfill rubbish but also open new markets of providing materials to potential manufactures, which contributes to business profit.

Domain-Dependent Sustainability: It should be noted that, as each domain has its own requirements, so sustainability has different manifestations for different domains. Sustainability for each domain must be analysed with different criteria. PetShop application, for example, is examined mostly under supply chain context which focuses on how to increase the sales, effectively exploit existing resources and extend the market. Therefore, besides basic function of selling products, the indirect criteria are improving customer satisfaction, reusing or reducing resources through all phases of e-business from viewing products to processing orders. Goals like customer convenience, optimised travel routes, new markets etc. become important sustainability concerns for an e-business. This list will inevitably be very different for a different study domain.

Metrics and Measurements: As discussed above, we utilised measurements via power consumption and processing time to quantify environmental impact of the software. This worked well in most scenarios, however, in a few cases we observed a conflict between different runs of the scenarios. For instance, in case of TC09 2 runs of the scenario measured to a significantly lower time than the other 3. It may be because some tasks depend of several 'outside' factors, such as network communications paths, loads on the DNS servers, etc. This issue needs to be further investigated. Additional work is required in evaluating the impact of the redesign on the software quality through additional metrics. We also observed that measurement of social sustainability impact was not feasible in this case study as our case study system was not in real use. However, the measurement process and metrics for this area of sustainability are also grossly under-researched at present.

VII. Conclusion

With continuous improvements in technology, more and more applications are developed to support human demands. However, when software is integrated into physical world, it affects many wider aspects of life (such as environment and society at large) which are very often not considered relevant during software development. This study reports on an experience of discovering and integrating such wider requirements into an existing application. Throughout the case study, we have observed a tension between different dimensions of sustainability. Thus, it is not sufficient to simply 'increase' sustainability of software, as this may very well mean that one sustainability dimension (e.g., economic) of that software system is very well supported, while others are neglected. For a system to be truly sustainable the increase in sustainability must also be achieved in balance, so that all pillars of sustainability are equally well supported. In the presented study, we have considered economists, environmentalist and socially concerned stakeholders, as we think that including such diverse stakeholders is a productive approach for capturing sustainability goals. Evaluation methods and metrics are also proposed for environmental and economic requirements. Here we convert performance of requirements into an amount of CO2 emissions, which is used as a common unit to measure sustainability. Overall, we have identified 19 sustainability requirements out of which which 11 were implemented, and the re-designed system was evaluated. Compared with the similar functions in the original version, the new version reduces CO2 emissions by 20% and increases processing time within the system by a factor of 28. As a future work, we are researching on factors that impact on e-commerce application in terms of the relationship between power consumption and processing time, deployment environment and data usage to achieve more accurate metrics to measure sustainability benefits, as well as problems in measuring social sustainability.

References

[1] C. Canon and M. Mahaux. Integrating the complexity of sustainability in requirements engineering. *First international workshop on Requirements for Sustainable Systems*, 2012.

[2] Eva Kern Markus Dick, Jakob Drangmeister and Stefan Naumann. Green software engineering with agile methods. In *GREENS*, pages 78–85. IEEE, 2013.

[3] J. Cabot, M. S. Easterbrook, J. Horkoff, L. Lessard, S. Liaskos, and N. J Mazón. Integrating sustainability in decision-making processes: A modelling strategy. *Software Engineering - Companion Volume, 2009. ICSE-Companion 2009. 31st International Conference on*, pages 207–210, 2009.

[4] M. Mahaux, P. Heymans, and G. Saval. Discovering sustainability requirements: An experience report. *Requirements Engineering: Foundation for Software Quality. 17th International Working Conference, REFSQ 2011*, pages 19–33, 2011.

[5] M. Berners-Lee. *How bad are bananas?* Canada : Greystone Books, 2011.

[6] S. Naumann, M. Dick, E. Kern, and T. Johann. The greensoft model: A reference model for green and sustainable software and its engineering. *Sustainable Computing: Informatics and Systems*, 1(4):294–304, 2011.

[7] F. L Steimer and T Steimer. Green e-business. *2010 4th IEEE International Conference on Digital Ecosystems and Technologies (DEST 2010)*, pages 252–253, 2010.

[8] M. Korte, K. Lee, and Chun Che Fung. Sustainability in information systems: Requirements and emerging technologies. *2012 International Conference on Innovation Management and Technology Research*, pages 481–485, 2012.

[9] L. Minas and B Ellison. The problem of power consumption in servers. http://lass.cs.umass.edu/~shenoy/courses/fall09/691gc/slides/Lec12-1-Slides-Power\%20Consumption\%20in\%20Servers_CS697GC_Fall2009-V02.ppt, Fall 2009. Accessed: 20/03/2013.

[10] Oracle and/or its affiliates the source for java technology collaboration. https://java.net/projects/petstore/sources/svn/show/tags/javapetstore-2_0-ea-v5_June_07?rev=699. Accessed: 20/12/2012.

[11] istarwiki.org istarquickguide. http://istar.rwth-aachen.de/tiki-index.php?page=iStarQuickGuide. Accessed: 20/03/2013.

[12] Mark Blackburn. Five ways to reduce data center server power consumption. White paper, The Green Grid, 2008.

[13] Spring Source Hyperic sigar - system information gatherer and reporter. https://support.hyperic.com/display/SIGAR/Home. Accessed: 20/12/2012.

[14] Mozilla Foundation(US) add-ons. https://addons.mozilla.org/en-US/firefox/addon/firebug/. Accessed: 12/05/2013.

[15] Energy saving power meter.